



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**“DISEÑO Y DESARROLLO DE UN SOFTPHONE
PARA TELEFONÍA IP UTILIZANDO EL PROTOCOLO IAX”**

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA

ALBERTO LAVARIEGA ARISTA

DIRECTOR DE TESIS

ING. HERIBERTO ILDEFONSO HERNÁNDEZ MARTÍNEZ

HUAJUAPAN DE LEÓN, OAX.; JUNIO DE 2007

Tesis presentada el 22 de junio de 2007 ante los siguientes sinodales:

M.C. David Martínez Torres

M.C. Jorge Arturo Hernández Perales

Dr. Antonio Orantes Molina

Director de Tesis:

Ing. Heriberto I. Hernández Martínez

Dedicatoria

Este trabajo de tesis está dedicado “a la mujer y al hombre del campo”, que con su esfuerzo, dedicación y el sudor de su frente han cultivado la tierra y generado la semilla de la humildad, la honradez y la moral, de ellos soy producto. Doy gracias a ese Ser que creó la vida sobre la faz de la tierra y nos enseñó que existe un único camino para alcanzar la felicidad.

A mi madre Ricarda, por brindarme su amor, bendiciones, consejos, confianza y sabiduría. A mi padre Manuel, por su amor, fortaleza, confianza y consejos, pero sobre todo por darme una lección de vida que nunca olvidaré en aquella etapa difícil en que pretendía abandonar mis estudios. Les agradezco infinitamente por ayudarme a ser el ser humano y el profesionalista que soy.

A mi hermano Fernando, por retarme y apoyarme a cumplir esta meta que parecía inalcanzable.

A mis hermanos, María, Abdón, Lorenzo, Bertha, Oliva y Janeth, por el apoyo incondicional brindado, y en especial a mi hermana Gregoria (Q.E.P.D) por ayudarme desde el cielo a cumplir con este objetivo *“Te quiero gollita, siempre estarás en mi corazón”*.

A mi amiga Karina Rosales A., por brindarme su amor y apoyo incondicional en los momentos difíciles; esperando que este trabajo le sirva de ejemplo para seguir adelante con sus metas.

“Mejorar cada día es mi meta, triunfar es mi lema y amar es la razón por la que estoy viviendo”.

“Cuando el viento, el tiempo y la distancia me hablan de ti, puedo saber que estás viva, sin dudar por un segundo que te amaré toda la vida”.

“No entiendo el porqué de mi existir, ni tampoco mi razón de ser; lo único que sé es que quiero ser feliz y eso resuelve todo el dilema de mi vida”.

Alberto.

Agradecimientos

Terminar este trabajo de tesis implicó tristezas, alegrías, esfuerzo y trabajo, pero no hubiera sido posible sin el apoyo de las personas que de una u otra manera me apoyaron.

A mi director de tesis Heriberto I. Hernández Martínez, por dirigir y brindar la experiencia necesaria, y que a pesar de mis tropiezos siempre estuvo ahí para apoyarme, brindarme sus consejos y alentarme para continuar con este trabajo de tesis.

A los sinodales, cDr. David Martínez Torres, M.C Jorge Arturo Hernández Perales y Dr. Antonio Orantes Molina, por sus aportaciones a este trabajo de tesis.

A la Universidad Tecnológica de la Mixteca, por brindarme el apoyo necesario durante la realización del presente trabajo.

A mi tío Celestino Lavariega Villalobos, por apoyarme en los inicios de mi carrera profesional, alentarme para no caer y demostrarme que *“el querer es poder”*.

Al Ing. Víctor Manuel Von Cruz, por su amistad y apoyo moral.

A los ingenieros Jehú López Aparicio, Josué Neftalí García Matías y José Manuel Avila Vázquez, por brindarme su apoyo técnico.

Y a todas aquellas personas que indirectamente me apoyaron para terminar este trabajo, por el apoyo moral incondicional que me han brindado.

Alberto.

Índice

Dedicatoria	v
Agradecimientos	vii
Índice	ix
Índice de tablas	xiii
Índice de figuras	xv
Resumen	xix
Abstract	xxi
Introducción	xxiii
Planteamiento del problema	xxv
Justificación del problema	xxviii
Objetivos	xxix
Estructura de la tesis	xxix
1. Estado del arte de las comunicaciones de voz sobre IP	1
1.1. Historia de la tecnología VoIP	3
1.2. Elementos de la tecnología VoIP	4
1.3. Características de la señal de voz sobre IP	4
1.3.1. Características de implementación de VoIP	5
1.3.1.1. Ancho de banda	5
1.3.1.2. Calidad en la transmisión de voz	6
1.3.1.3. Estándarización	6
1.3.1.4. Regulaciones gubernamentales	6
1.3.1.5. Principales aplicaciones de VoIP	7
1.3.2. Ventajas y desventajas sobre el uso de VoIP	7
1.4. Razones que fundamentan el uso de VoIP	9
1.4.1. Caso comercial	9
1.4.2. Presencia universal de IP	9
1.4.3. Maduración de las tecnologías	10
1.4.4. Migración a redes de datos	10
1.5. Codecs de voz	11

1.5.1. Funciones de los codecs.....	11
1.5.2. Clasificación de los codecs.....	11
1.5.3. Evaluación de codecs.....	12
1.5.4. Comparativa de codecs.....	12
1.5.4.1. ILBC.....	12
1.5.4.2. g.729a.....	12
1.5.4.3. g.711 (u-law y a-law).....	13
1.5.4.4. GSM.....	13
1.5.4.5. Speex.....	13
1.6. Modelo VoIP.....	16
1.6.1. Transmisión de voz sobre IP.....	16
1.6.2. Transmisión de voz sobre UDP.....	16
1.6.3. Transmisión de voz sobre RTP.....	16
1.6.4. La transmisión de voz sobre TCP no es recomendable.....	17
1.6.5. VoIP en Web.....	17
1.6.6. Protocolos VoIP agrupados en planes.....	17
2. Protocolos de establecimiento de llamada.....	19
2.1. H.323.....	20
2.1.1. Consideraciones de seguridad.....	21
2.1.2. H.323 y NAT.....	21
2.1.3. Arquitectura de H.323.....	21
2.1.4. Operaciones H.323.....	22
2.1.5. Descripción general de H.323.....	24
2.2. MGCP.....	25
2.2.1. Arquitectura MGCP.....	26
2.2.2. Operaciones MGCP.....	26
2.3. SIP.....	29
2.3.1. Consideraciones de seguridad.....	30
2.3.2. SIP y NAT.....	30
2.3.3. Arquitectura SIP.....	31
2.3.4. Operaciones SIP.....	32
2.4. Protocolo IAX.....	33
2.4.1. IAX y NAT.....	33
2.4.2. Descripción del protocolo IAX.....	34
2.4.3. Indicador uniforme del recurso IAX.....	34
2.4.4. Transporte de mensajes IAX.....	35
2.4.4.1. Registro.....	35
2.4.4.2. Administrador de enlaces de llamada.....	37
2.4.4.3. Control de llamada.....	37
2.4.4.4. Operaciones de conexión entre llamada.....	38

2.4.4.5. Optimización de la trayectoria de llamada	38
2.4.4.6. Finalización de llamada	39
2.4.4.7. Supervisión o monitoreo de la red	39
2.4.4.8. Dígito de marcado	40
2.4.4.9. Descarga de firmware	40
2.4.4.10. Aprovisionamiento	41
2.4.4.11. Misceláneos	42
2.4.4.12. Mensajes media	42
2.4.5. Codificación de mensajes	46
2.4.5.1. Tramas F	46
2.4.5.2. Tramas M	47
2.4.5.3. Meta tramas	47
2.4.6. Tipos de trama	49
2.4.7. Elementos de información	50
2.4.8. Encriptado	50
2.4.9. Consideraciones de seguridad	50
3. Desarrollo de la aplicación <i>softphone XoIP</i>	53
3.1. Planeación inicial	53
3.1.1. Herramientas software involucradas	53
3.1.1.1. Rational Rose	53
3.1.1.2. QT Designer	53
3.1.1.3. Asterisk	54
3.2. Requisitos del <i>softphone XoIP</i>	54
3.2.1. Evaluación de <i>softphones</i> existentes en el mercado	54
3.2.2. Actores y casos de uso	57
3.2.3. Priorizar casos de uso	57
3.2.4. Función de los casos de uso del <i>softphone XoIP</i>	58
3.3. Análisis y diseño	60
3.3.1. Análisis	61
3.3.1.1. Análisis de la arquitectura	61
3.3.1.2. Análisis de los casos de uso	61
3.3.1.3. Análisis de clases	62
3.3.2. Diseño	63
3.3.2.1. Diseño de la arquitectura y los subsistemas	63
3.3.2.2. Diseño de casos de uso	63
3.3.2.3. Diseño de clases	64
3.3.2.4. Identificación de los objetos	64
3.3.3. Especificación de los casos de uso	64
3.3.3.1. Especificación del caso de uso Crear cuenta	65
3.3.3.2. Especificación de caso de uso Iniciar sesión	67

3.3.3.3. Especificación de caso de uso Realizar llamada telefónica.....	71
3.3.3.4. Especificación de caso de uso Cerrar sesión	74
3.4. Implementación	78
3.4.1. Reuso en la implementación.....	79
3.4.2. Implementación de las clases de diseño	79
3.4.3. Asterisk y el <i>softphone XoIP</i>	81
3.5. Pruebas del <i>softphone XoIP</i>	81
4. Resultados.....	85
5. Conclusiones y líneas futuras de investigación	89
6. Bibliografía.....	91
Lista de acrónimos.....	93
Anexo A. Manual de usuario del <i>softphone XoIP</i>	A-1

Índice de tablas

Tabla 1.1. Comparativa de codecs.....	14
Tabla 1.2. Soporte Cisco de codecs de la ITU-T.....	15
Tabla 2.1. Funciones y protocolos que utiliza H.323.....	21
Tabla 2.2. Mensajes utilizados para el registro del par.....	36
Tabla 2.3. Mensajes utilizados en la administración de la llamada.....	37
Tabla 2.4. Mensajes utilizados en el control de llamada.....	38
Tabla 2.5. Mensajes utilizados en las operaciones de conexión entre llamada.....	38
Tabla 2.6. Mensajes utilizados en la optimización de la trayectoria de la llamada.....	38
Tabla 2.7. Mensajes IAX para el monitoreo de la red.....	39
Tabla 2.8. Mensajes IAX para marcación.....	40
Tabla 2.9. Mensajes IAX utilizados en la descarga de controladores <i>firmware</i>	40
Tabla 2.10. Mensaje de aprovisionamiento.....	41
Tabla 2.11. Mensajes misceláneos.....	42
Tabla 2.12. Mensajes media IAX.....	42
Tabla 2.13. Tipos de tramas especificadas por el protocolo IAX.....	49
Tabla 2.14. Elementos de información.....	51
Tabla 3.1. Comparativa de características técnicas de los <i>softphones</i>	55
Tabla 3.2. Comparativa detallada de <i>softphones</i>	56
Tabla 3.3. Requisitos del <i>softphone XoIP</i>	58
Tabla 3.4. Identificación de los estereotipos de las clases.....	62
Tabla 3.5. Historial de revisiones del caso de uso Crear cuenta.....	65
Tabla 3.6. Flujo básico del caso de uso Crear cuenta.....	65
Tabla 3.7. Historial de revisiones del caso de uso Iniciar sesión.....	68
Tabla 3.8. Flujo básico del caso de uso Iniciar sesión.....	68
Tabla 3.9. Historial de revisiones del caso de uso Realizar llamada telefónica.....	71
Tabla 3.10. Flujo básico del caso de uso Realizar llamada telefónica.....	72
Tabla 3.11. Historial de revisiones del caso de uso Cerrar sesión.....	74
Tabla 3.12. Flujo básico del caso de uso Cerrar Sesión.....	74

Índice de figuras

Figura 1.1. Comunicación VoIP entre sucursales.....	2
Figura 1.2. Videoconferencia.	2
Figura 1.3. Elementos principales en la transmisión de VoIP.....	4
Figura 1.4. Configuración típica para la transmisión de voz a través de Telefonía IP.....	5
Figura 1.5. Localización de IP vs ATM y Frame Relay.....	10
Figura 1.6. Descripción de ATM en la red.....	10
Figura 1.7. Conjunto de protocolos VoIP correspondientes a la familia de protocolos de Internet...	16
Figura 1.8. VoIP basada en Web.....	17
Figura 1.9. Modelo basado en Web agrupado en tres planes.....	18
Figura 2.1. Procedimiento de establecimiento de llamada VoIP.....	20
Figura 2.2. Arquitectura de H.323.....	22
Figura 2.3. Operaciones de H.323.....	22
Figura 2.4. Operaciones de descubrimiento del <i>gatekeeper</i>	23
Figura 2.5. Procedimientos de registro.....	24
Figura 2.6. Admisión de procedimientos.....	25
Figura 2.7. Arquitectura MGCP.....	26
Figura 2.8. Flujos de eventos del protocolo MGCP.....	27
Figura 2.9. Clientes y servidores SIP.....	32
Figura 2.10. Operaciones SIP.....	32
Figura 2.11. Secuencia de solicitud de registro IAX.....	36
Figura 2.12. Secuencia de liberación del registro IAX.....	37
Figura 2.13. Secuencia para la optimización de la trayectoria de la llamada.....	39
Figura 2.14. Secuencia de mensajes para monitorear la red y para determinar el retraso entre dos pares.....	40
Figura 2.15. Secuencia de mensajes para descargar los controladores <i>firmware</i>	41
Figura 2.16. Secuencia de mensajes para el aprovisionamiento de un dispositivo IAX.....	41
Figura 2.17. Secuencia de mensajes para la transmisión de mensajes <i>media</i>	43
Figura 2.18. Secuencia de mensajes para la transmisión de mensajes <i>media</i> vía dispositivos.....	45
Figura 2.19. Formato binario de una trama F.....	46

Figura 2.20. Formato binario de una trama M.....	47
Figura 2.21. Formato binario de una meta trama de video.	47
Figura 2.22. Formato binario de la meta trama <i>trunk</i> (sello de tiempo = 0).....	48
Figura 2.23. Formato binario de la meta trama <i>trunk</i> (sello de tiempo = 1).....	49
Figura 2.24. Formato binario de un IE.	50
Figura 3.1. Diagrama general de casos de uso del <i>softphone</i> XoIP.....	59
Figura 3.2. Modelo de capas típico de XoIP.	62
Figura 3.3. Diagrama de actividades del caso de uso Crear cuenta.....	66
Figura 3.4. Diagrama de secuencia del caso de uso Crear cuenta.	66
Figura 3.5. Diagrama de colaboración del caso de uso Crear cuenta.....	67
Figura 3.6. Diagrama de estado del caso de uso Crear cuenta.....	67
Figura 3.7. Diagrama de clases del caso de uso: Crear cuenta.	68
Figura 3.8. Diagrama de actividades del caso de uso Iniciar sesión.....	69
Figura 3.9. Diagrama de secuencia del caso de uso Iniciar sesión.....	69
Figura 3.10. Diagrama de colaboración del caso de uso Iniciar sesión.....	70
Figura 3.11. Diagrama de estado del caso de uso Iniciar sesión.....	70
Figura 3.12. Diagrama de clases del caso de uso Iniciar sesión.....	71
Figura 3.13. Diagrama de actividades del caso de uso Realizar llamada telefónica.....	72
Figura 3.14. Diagrama de secuencia del caso de uso Realizar llamada telefónica.....	73
Figura 3.15. Diagrama de colaboración del caso de uso Realizar llamada telefónica.....	73
Figura 3.16. Diagrama de estado del caso de uso Realizar llamada telefónica.....	74
Figura 3.17. Diagrama de clases del caso de uso Realizar llamada telefónica.....	75
Figura 3.18. Diagrama de actividades del caso de uso Cerrar sesión.....	76
Figura 3.19. Diagrama de secuencia del caso de uso Cerrar sesión.....	76
Figura 3.20. Diagrama de colaboración del caso de uso Cerrar sesión.....	77
Figura 3.21. Diagrama de estado del caso de uso Cerrar sesión.....	77
Figura 3.22. Diagrama de clases del caso de uso Cerrar sesión.....	78
Figura 3.23. Ejemplo de reuso de la clase QEvent de QT Designer y de la librería IAXclient.....	79
Figura 3.24. Evolución de las clases <i>Entity</i> de diseño en componentes de implementación.....	79
Figura 3.25. Evolución de las clases <i>Boundary</i> de diseño en componentes de implementación.....	80
Figura 3.26. Evolución de las clases <i>Control</i> de diseño en componentes de implementación.....	80
Figura 3.27. Configuración de los archivos asterik.conf y extensions.conf.....	81
Figura 3.28. Diagrama de instalación y configuración del Servidor/IP PBX.....	82
Figura 3.29. Archivo de configuración sip.conf.....	83
Figura 3.30. Archivo de configuración iax.conf.....	83
Figura 3.31. Servidor/IP PBX en ejecución.....	84
Figura 4.1. Diagrama de flujo general sobre el funcionamiento de XoIP.....	85
Figura 4.2. Cliente XoIP registrado en el Servidor/IP PBX.....	86
Figura 4.3. Establecimiento de llamada con otro cliente IAX registrado en el Servidor/IP PBX.....	86
Figura 4.4. Aplicación que usa el protocolo SIP y que se encuentra registrada en el Servidor/IP PBX.....	

.....	87
Figura 4.5. Establecimiento de llamada de un cliente SIP con un cliente IAX.....	87
Figura 4.6. Teléfono IP registrado en el Servidor/IP PBX.....	88
Figura 4.7. Adaptador telefónico utilizado como puente para el registro con el Servidor IP/PX.....	88

Resumen

A partir de la década de los noventa, el campo de las telecomunicaciones de voz ha venido desarrollando nuevas tecnologías, una de ellas es la tecnología de voz sobre el protocolo IP (VoIP) en donde la principal innovación es la implementación de un teléfono en software, conocido como *softphone*. En la actualidad existe una gran variedad de soluciones, las soluciones propietarias representan costos de funcionamiento, mientras que la mayoría de soluciones libres no proporcionan la totalidad de las funciones de un teléfono analógico.

El presente trabajo de tesis plantea el diseño e implementación de un *softphone* basado en el protocolo de establecimiento de llamadas IAX para cumplir con los requerimientos de la tecnología VoIP y ser una solución abierta a las necesidades de comunicación dentro de las instalaciones de la Universidad Tecnológica de la Mixteca.

Durante el ciclo de vida del *softphone XoIP* se emplean las técnicas del proceso unificado (UP) y el lenguaje de modelado UML con la finalidad de obtener un software de calidad, asimismo, la implementación del *softphone XoIP* se basó en la reutilización de componentes.

Se describe la tecnología VoIP y el protocolo de establecimiento de llamadas IAX; posteriormente se detalla el diseño y desarrollo del *softphone XoIP* con base en los flujos de trabajo fundamentales que especifica el UP. Finalmente se presentan los resultados, un manual de usuario de la herramienta final y las conclusiones correspondientes.

Abstract

From the 1990s the field of voice telecommunications has been developing new technologies. One of these is voice over IP (VoIP) where the principal innovation is the implementation of a telephone in software, known as softphone. Nowadays, there is a great variety of solutions. Proprietor solutions represent functioning costs, while the majority of free solutions do not provide the all the functions of an analog telephone.

The present study considers the design and implementation of a softphone based on the IAX protocol to fulfill the requirements of VoIP technology and be an open solution to the communication needs in the Technological University of the Mixteca.

During the life cycle of the softphone XoIP unified process (UP) is employed as well as Unified Model Language (UML) with a view to obtaining quality software. Likewise, the implementation of the softphone XoIP was based on the reusability of components.

VoIP technology and the IAX protocol are described. Later, the design and development of softphone XoIP is described in detail based on the fundamental workflows that specify the UP. Finally, results are presented together with a user's manual for the final software and the corresponding conclusions

Introducción

Actualmente, la mayoría de empresas utilizan una línea telefónica como medio de comunicación, la cual está administrada por alguna empresa como Telmex, Avantel, Nextel, Telcel, Iusacell, etc. Dicho medio de comunicación tuvo sus inicios hace aproximadamente 23 años, durante los cuales el mundo de las telecomunicaciones ha venido a ser uno de los mercados de mayor ingreso económico y de alto nivel tecnológico. Sin embargo, los altos costos que representan la tecnología y servicios adicionales están ocasionando que los usuarios finales de telefonía busquen soluciones más baratas.

A partir de 1995, expertos en informática, electrónica y tecnologías de la información (TI) enfocaron sus esfuerzos a incorporar la voz en formato digital en comunicaciones de datos (Internet) considerando los avances en tecnología de redes y la capacidad de adopción de nuevas tecnologías del propio Internet.

Las conversaciones telefónicas mediante redes de datos se han venido incrementando, principalmente utilizando la transmisión de voz sobre IP (VoIP, *Voice over Internet Protocol*), a pesar de que inicialmente la calidad de comunicación fue considerada nefasta [3].

En 1995 se había desarrollado un teléfono implementado en software, conocido como *soft-phone*¹, y que permitía llamadas de voz entre computadoras personales (PC, *Personal Computer*); actualmente existen tecnologías que contribuyen al crecimiento y permiten entablar comunicación usando las siguientes configuraciones: PC a PC, PC a teléfono (móvil o fijo) y teléfono a PC.

La configuración PC a teléfono (movil o fijo) motivó a los desarrolladores de VoIP a implementar calidad de servicio (QoS, *Quality of Service*) en las llamadas telefónicas mediante Internet, lo que implicó eliminar eco y ruido, suprimir silencios², evitar la pérdida de paquetes y el retardo en su envío.

En 2005, la telefonía IP logró la etapa de adopción, proporcionando un ahorro aproximado del 80% en costos de comunicaciones de voz en empresas grandes, medianas y pequeñas [URL18]. Cabe señalar que las empresas más beneficiadas han sido aquellas que cuentan con diversas sucursales, ya que sus llamadas internas e intersucursales no representan costo alguno por estar entrelazadas mediante una red de datos.

Las estadísticas indican que durante el 2006, gran parte de las llamadas telefónicas se realizaron a través de Internet.

¹ *Softphone* es el nombre que se le da al teléfono implementado en software.

² En la técnica de supresión de silencios, sólo se envían los paquetes de voz digitalizados cuando se detecta que existe actividad de voz, de lo contrario no se envía nada y con ello se libera el ancho de banda.

Respecto a las empresas de telefonía convencional, éstas empiezan a considerar las grandes ventajas que representa la telefonía IP, por lo tanto intentan adoptarla como parte de sus servicios a pesar de que dicha adopción implica altos costos y cambios en su infraestructura.

La VoIP inicialmente se implementó para reducir el ancho de banda mediante compresión de voz, aprovechando los procesos de compresión implementados en los sistemas celulares de la década de los 80s, con lo cual se redujeron los costos en el transporte internacional. Posteriormente, se aplicó en la red digital de servicios integrados (RDSI) sobre redes de área local (LAN, *Local Area Network*) e Internet, y finalmente se migró a aplicaciones privadas sobre LANs y aplicaciones públicas sobre redes de área amplia (WAN, *Wide Area Network*).

Actualmente, el crecimiento de la tecnología VoIP ha sido significativo, al grado de que tanto empresas como usuarios particulares hacen uso de ella. El cambio de servicio, telefonía convencional a VoIP, se debe principalmente a que los costos de comunicación de los actuales operadores de telefonía (*carriers*) son elevados, debido a ello las empresas y usuarios empezaron a buscar la manera de abaratar costos. Las opciones no fueron muchas, así que la telefonía IP y VoIP fueron las únicas opciones, logrando un ahorro en costos de llamadas telefónicas de hasta el 80% mediante el uso de Internet público como medio de transmisión.

Uno de los primeros operadores de telefonía IP fue Skype, empresa fundada por Niklas Zennström y Janus Friis³; el *softphone* que proporcionan cumple con las funciones de un teléfono analógico para el establecimiento de llamadas de voz, las cuales son gratuitas al realizarse entre PCs. Skype cuenta con los siguientes servicios [URL13]:

- *SkypeOut*: Permite llamar a teléfonos convencionales mediante el pago de una tarifa.
- *SkypeIn*: Permite recibir llamadas de teléfonos convencionales mediante el pago de una tarifa.

Otros operadores son Vonage, net2Phone, Sitatel, Ubifone y PeopleCall, los cuales se diferencian por la calidad y las tarifas de las llamadas a teléfonos convencionales. El uso de VoIP se ha extendido a una gran diversidad de aplicaciones, por ejemplo los mensajeros instantáneos (Messenger, Yahoo Messenger, Google Talk, etc.).

Las herramientas, software y hardware, y los servicios que ofrece la tecnología VoIP han hecho que las grandes empresas de las telecomunicaciones consideren a la telefonía IP como una seria competencia; para ello intentan ofrecer servicios similares con la finalidad de mantener su cartera de clientes, sin embargo la competencia es cada vez más grande pues existen firmas de gran renombre como Cisco y Avaya que ofrecen equipos y soluciones con tecnología VoIP.

Otra empresa que apoya el desarrollo de la tecnología VoIP es CounterPath, la cual diseña y desarrolla *softphones* para venderlos como licencias; sus *softphones* más conocidos son eyeBeam, Pocket PC y X-PRO & X-Lite, y sus principales consumidores son las empresas Ubifone y Ineen Phone [URL20].

Gracias al avance de la tecnología VoIP, el desarrollo y venta de *softphones* es uno de los mercados con mayor potencial dentro del campo de las telecomunicaciones por IP, debido principalmente a que el *softphone* convierte a la computadora en un teléfono.

El *softphone*, como elemento de la telefonía IP, es la parte innovadora ya que el uso de los actuales teléfonos IP no viene a favorecer al concepto de telefonía por Internet, sin embargo el uso de un *softphone* provoca especial interés por el uso de su tecnología, y sobre todo por los bajos costos de llamada que presenta. El *softphone* es parte importante en el crecimiento del mercado actual de las telecomunicaciones por Internet.

³ Creadores del software Kazaa para conexiones par a par o P2P (*Peer-to-Peer*).

En la actualidad existe una gran diversidad de *softphones*, los cuales son desarrollados por empresas particulares o usuarios independientes, que se caracterizan por su funcionalidad, soporte a diferentes codecs y protocolos de establecimiento de llamada. Cabe señalar que los protocolos de establecimiento de llamadas más importantes, por su uso y funcionalidad, son SIP, IAX y H.323 [3].

Los *softphones* más populares son X-PRO & Xlite, eyeBeam de la firma CounterPath, Skype, Gizmo, etc., de los cuales Xlite es el único no comercial y por ello no cuenta con todas las funcionalidades que incluye un *softphone* propietario.

Los *softphones* gratuitos (*freeware*⁴) y de código abierto (*open source*) más conocidos son ohPhone y OpenPhone, desarrollados por el OpenH323 Project [URL11], los cuales no poseen una interfaz gráfica de usuario (GUI, *Graphics User Interface*) amigable y sus características de uso no son transparentes a los usuarios, que generalmente desconocen de la tecnología VoIP. También se encuentran SiphezPhone y SipxPhone de la firma SipFoundry [URL12], *softphones* de código abierto que carecen de muchas funcionalidades y presentan ciertos conflictos de interconexión con servidores de voz cuando éstos se encuentran detrás de NAT (*Network Address Translation*)⁵ o un cortafuegos (*firewall*)⁶ debido a la utilización del estándar SIP. Ejemplos de *softphones* que soportan el protocolo de establecimiento de llamadas IAX son IaxClientOcx, ZiaxPhone y IDEFisk, desarrollados a criterio de un grupo independiente de desarrolladores.

Planteamiento del problema

En la actualidad existen varias aplicaciones comerciales de *softphone* para telefonía IP, algunas de ellas son gratuitas pero no están documentadas y su funcionalidad no es completamente segura debido a que son realizadas con fines de prueba antes de lanzarlos al mercado como soluciones propietarias; algunas otras son funcionales pero no cumplen con las características de un *softphone* como son transferencia de llamadas, uso de DTMF (*Dual Tone MultiFrequency*) para realizar llamadas o consultar el buzón de voz, historial de llamadas recibidas y realizadas, identificador de llamadas, etc.

En la Universidad Tecnológica de la Mixteca (UTM) no existe trabajo alguno vinculado a la investigación y desarrollo de telefonía IP, sin embargo el departamento de red ofrece una solución propietaria de telefonía VoIP, cuyo uso es totalmente transparente a los usuarios. El presente trabajo de tesis plantea el diseño e implementación del *softphone XoIP*, que cumpla los requerimientos de la tecnología VoIP y sea una solución abierta a las necesidades de comunicación dentro de las instalaciones de la UTM.

El desarrollo del *softphone XoIP* emplea el paradigma de la programación orientada a objetos (OOP, *Object Oriented Programming*), el cual ofrece técnicas y métodos para desarrollar software de calidad e implica la creación de objetos del mundo real. Este paradigma hace énfasis en el análisis y el diseño con el principal objetivo de lograr, en la mayor medida, la reutilización tanto del código como del proceso de desarrollo. Otras de las características a considerar de la OOP es que lleva a menos efectos colaterales cuando es necesario hacer modificaciones y proporciona mayor adaptabilidad y capacidad de escalabilidad [9, 13, 15, 16].

⁴ *Freeware* es un software que se distribuye sin costo. En algunas ocasiones se incluye el código fuente.

⁵ NAT es un método que permite conectar múltiples computadores a Internet o a cualquier otro tipo de red IP utilizando únicamente una dirección IP o un número reducido de ellas.

⁶ Un cortafuegos en Internet es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de una red privada (VPN, *Virtual Private Network*) y el Internet. El cortafuegos determina los servicios de red que pueden ser accedidos desde el exterior, es decir quién puede entrar para utilizar los recursos de red pertenecientes a la organización.



Figura I.1. Flujos de trabajo fundamentales del UP.

En la actualidad las empresas de desarrollo de software tienden a crear software más grande y más complejo, además, los clientes tienden a ser más exigentes en cuanto a calidad, tiempo, eficacia y eficiencia del software a entregar, esto ha llevado a que el análisis y diseño de los sistemas software se lleven a cabo bajo un proceso riguroso [16]. Por lo anterior, durante el ciclo de vida del *softphone* XoIP se emplean las técnicas del proceso unificado (UP, *Unified Process*) y el lenguaje de modelado UML (*Unified Model Language*). El UP es un proceso de desarrollo de software definido como “el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software” [11].

Las características que definen al UP son: está dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental; además está dividido en cuatro fases: inicio, elaboración, construcción y transición. Estas fases son utilizadas durante el tiempo de vida del software, cada una de las cuales se divide en un conjunto de iteraciones que se repiten a lo largo de una serie de ciclos, donde al término de cada ciclo se obtiene una versión del software listo para ser entregado al cliente. Una iteración puede pasar por los cinco flujos de trabajo fundamentales del UP: requisitos, análisis, diseño, implementación y pruebas (Figura I.1) [12, 13].

Se puede decir que el UP es un proceso de ingeniería de software que tiene como finalidad proporcionar una disciplina para la asignación de tareas y responsabilidades dentro de una organización de desarrollo, con el objetivo de asegurar la producción de software de calidad que satisfaga las necesidades de los usuarios finales dentro de un tiempo y presupuesto visible [10, 16].

En cuanto al lenguaje de modelado, se utilizó el UML por ser un estándar en el análisis y diseño de sistemas software. UML es un lenguaje de modelado gráfico y visual utilizado para especificar, visualizar, construir y documentar los componentes de un sistema software. Está pensado para utilizarse en cualquier proyecto de desarrollo que necesite capturar requerimientos y comportamientos del sistema que se desee construir; ayuda a comprender y mantener de una mejor forma un sistema, que un analista de sistemas o desarrollador pueda desconocer [13].

UML permite captar la información sobre la estructura estática y dinámica de un sistema, en donde la estructura estática proporciona información sobre los objetos que intervienen en determinado proceso y las relaciones que existen entre ellos; la estructura dinámica define el comportamiento de los objetos a lo largo de todo el tiempo que interactúan hasta llegar a cumplir con sus objetivos [17].

Por otro lado, el concepto de reutilización se dio a conocer en 1968 durante la conferencia de la OTAN sobre ingeniería de software en la que Doug McIlroy abogó por la “producción en masa de componentes de software”. Sin embargo, fue a través del tiempo como se demostró que lo que Doug McIlroy mencionó no fue sólo un sueño, ahora es una realidad con la que se ha logrado reducir el costo de mantenimiento y de desarrollo de software [14, 15].

Fue con la OOP con la que se empezó a dar especial interés a la palabra “reutilización”, pues con el concepto de objeto se prometía proporcionar la reusabilidad que se deseaba, sin embargo la OOP no fue tan eficaz como se esperaba en cuestión de reutilización y ante esta situación surgió un nuevo paradigma, la programación orientada a componentes (OPC, *Oriented Programming Components*), que comparada con las clases o prototipos de objetos, los componentes ofrecen ciertos requisitos, como por ejemplo la posibilidad de utilizarlos con independencia de otros elementos [8].

Es por ello que durante el desarrollo de un sistema software se recomienda la reutilización a gran escala con la finalidad de esperar los siguientes beneficios [14]:

- *Oportunidad*: Al basarse en componentes ya existentes se tiene la ventaja de desarrollar menos software y por lo tanto reducir el tiempo de desarrollo.
- *Disminución de los esfuerzos de mantenimiento*: Cuando se está a cargo de un sistema software, se debe pensar en la evolución del sistema. Esto lleva a que el sistema tenga nuevas funcionalidades, adaptaciones a nuevas plataformas, corrección de errores, etc. Lo que representa más trabajo, sin embargo el hacer uso del concepto reutilización ahorra trabajo y esfuerzo al momento de dar mantenimiento a un sistema software.
- *Fiabilidad*: Al basarse en componentes software de fuentes bien establecidas, se tiene la garantía, o al menos la expectativa, de que sus autores hayan aplicado todo el cuidado necesario incluyendo comprobaciones y otras técnicas de verificación; por no mencionar la esperanza de que otros desarrolladores hayan tenido la oportunidad de tratar esos componentes anteriormente subsanando los fallos restantes.
- *Eficiencia*: Los mismos factores que favorecen la reutilización incitan a los desarrolladores de componentes a usar los mejores algoritmos y estructuras de datos que conozcan en su campo de especialización, mientras que en un gran proyecto de aplicación difícilmente se puede contar con un experto para cada campo.
- *Consistencia*: No hay una buena biblioteca sin un énfasis estricto en un diseño regular y coherente. Si se comienza por utilizar una biblioteca, en particular alguna de las mejores bibliotecas orientadas a objetos, su estilo comenzará a influir a través de un proceso natural de ósmosis, en el estilo del software que se desarrollará. Esto es un gran estímulo para la calidad del software producido por un grupo de aplicación.
- *Inversión*: Hacer el software reutilizable es una forma de preservar el *saber hacer* (*know-how*) y las invenciones de los mejores desarrolladores; transformando un recurso frágil en un valor permanente.

Con todos estos beneficios y las ventajas de la reutilización, se coincide con la idea de que el reuso es el proceso de crear software a través de software existente en lugar de construir software desde la nada [15].

La reutilización viene a simplificar el trabajo de calidad a los desarrolladores de software, pero la idea principal no es ésta, la idea va enfocada a que se convierta en un estándar, con la finalidad de que siempre que se deseen desarrollar sistemas software se considere la reutilización como primera instancia. Por tal motivo se han creado técnicas o aproximaciones que puedan ser utilizadas en el reuso de software; éstas son técnicas que de una u otra manera ayudan a los desarrolladores de software a entender la técnica que están utilizando o vayan a utilizar:

- *Aproximación tecnológica*: Se refiere a la técnica utilizada para desarrollar componentes.
- *Enfoque metodológico*: Representa la manera como se integra la reutilización dentro del proceso mismo de desarrollo.
- *Modificación*: Sugiere la forma en cómo los componentes son accedidos y manipulados para adecuarlos a nuevas necesidades.

- *Alcance de desarrollo*: Indica si la reutilización sólo se aplica a componentes internos o permite integrar componentes de otras bibliotecas.
- *Alcance del dominio*: Delimita la reutilización a una familia de sistemas o permite su aplicación entre diferentes dominios de aplicación.
- *Objeto de reuso*: Se refiere a los diferentes enfoques que diversos autores proponen al considerar la reutilización, desde el punto de vista del proceso de desarrollo, considerando el nivel de abstracción del componente, o según el tipo de conocimiento que se reutiliza.

La Tabla I.1 presenta una visión general del tratamiento de la reutilización considerando diferentes aspectos [1].

Tabla I.1. Técnicas de reutilización de software.

Técnica	Tipo	Descripción
Aproximación tecnológica	Por generación o adaptativo	Parte de una especificación del problema que va siendo refinada en iteraciones sucesivas
	Por composición	Utiliza pequeñas partes de software como invariantes para proveer funcionalidad variante por medio de su integración
Enfoque metodológico	Desarrollo para reutilizar	Centrado en el desarrollo de componentes adecuados para ser utilizados en un problema específico (visión proveedor)
	Desarrollo con reutilización	Centrado en el proceso mismo de construcción de soluciones software a partir de componentes almacenadas en una biblioteca (visión demanda)
Modificación	Caja blanca	Componentes que son reutilizados por modificación y adaptación
	Caja negra	Componentes que son reutilizados sin ninguna modificación
	Adaptativo	Utiliza estructuras grandes de software como invariantes y restringe la variabilidad a un conjunto de argumentos o parámetros
Alcance del desarrollo	Interno	Mide el nivel de reutilización de componentes de un repositorio interno o de componentes del mismo proyecto
	Externo	Mide el nivel de reutilización de componentes que provienen de repositorios externos o la proporción de productos que fueron adquiridos
Alcance del dominio	Vertical	Reutilización dentro del mismo dominio de aplicación
	Horizontal	Reutilización dentro de dominios de aplicación diferentes
Objeto de Reuso	Según el estado del proceso de desarrollo	Según la etapa en el ciclo de vida en la cual el conocimiento es producido o reutilizado. Tecnologías de reuso de amplio espectro y de limitado espectro [2]
	Según el nivel de abstracción	Código reciclado, componentes de código, esquemas Código fuente, diseño, especificación, objetos, texto, arquitecturas
	Según la naturaleza del conocimiento	Artefactos reutilizables: datos, arquitecturas, diseño, programas Naturaleza del conocimiento: informal, esquematizado, herramientas y productos

Justificación del problema

El uso de un *softphone* es indispensable dentro de un sistema de telefonía IP, ya que satisface las necesidades de movilidad y de acceso al conmutador telefónico de una empresa o institución, sin importar la distancia o el lugar en el que se encuentre el usuario; la única condición es contar con acceso a Internet para cumplir con las expectativas de comunicación.

El uso de teléfonos IP (*hardphone*⁷) es favorable, siempre y cuando se utilicen dentro de las instalaciones de la empresa o institución, sin embargo su utilización de forma remota es incómoda por cuestiones de transportabilidad.

El *softphone* es un elemento de gran utilidad para usuarios que continuamente están viajando y que necesitan acceder a su buzón de voz o llamar a la empresa o institución para la cual trabajan; mediante telefonía IP ahorran en costos de comunicación. Con el *softphone*, el usuario utiliza las ventajas que ofrecen las PCs, además de evitar los costos que representan los teléfonos convencionales y/o teléfonos IP.

Por otro lado, muchas empresas que cuentan con un sistema de telefonía IP dependen totalmente de la empresa que proporciona el servicio, debido a que no cuentan con personal especializado que pueda solucionar los problemas que dicha tecnología presenta. Por lo anterior, es recomendable que la adopción de la tecnología VoIP considere la especialización de su personal en esta nueva tecnología.

Objetivos

El objetivo principal de este trabajo de tesis es diseñar y desarrollar el *softphone XoIP* para telefonía IP mediante el protocolo de establecimiento de llamadas IAX, para ello es necesario hacer un estudio detallado de la tecnología VoIP con la finalidad de desarrollar un *softphone* de uso transparente y de documentación sencilla.

Para cumplir el objetivo principal, se plantean los siguientes objetivos secundarios:

- Analizar los elementos involucrados en los sistemas de telecomunicaciones sobre IP.
- Investigar los diferentes codecs existentes.
- Investigar los principales protocolos de establecimiento de llamada empleados en tecnología VoIP.
- Configurar y poner en funcionamiento una centralita VoIP (Servidor/IP PBX, *Internet Protocol - Private Branch Exchange*) para dar soporte a *softphones* con tecnología SIP o IAX.
- Probar el funcionamiento de diversos *softphones* existentes, utilizando el Servidor/IP PBX.
- Modelar el diseño del *softphone XoIP* mediante el UML de acuerdo a los requerimientos establecidos.
- Diseñar la GUI del *softphone XoIP*.
- Documentar el desarrollo del *softphone XoIP* mediante la utilización de herramientas de modelado y realizar un documento que acompañe a la aplicación final, para que el usuario obtenga los conocimientos teóricos y técnicos sobre el funcionamiento de la herramienta desarrollada.

La hipótesis que se plantea, es que con el diseño, modelado y desarrollo del *softphone XoIP*, los usuarios y desarrolladores de software VoIP conocerán el funcionamiento e identificarán los elementos involucrados en los sistemas de telecomunicaciones por IP.

Estructura de la tesis

La presentación del trabajo realizado se estructura como muestra la Figura I.2 y se detalla de la siguiente manera:

⁷ Al teléfono implementado en hardware se le conoce como teléfono IP o *hardphone*.

El Capítulo 1 presenta un estado del arte de la tecnología VoIP.

Los protocolos de establecimiento de llamada se describen en el Capítulo 2, poniendo especial atención en el protocolo IAX.

El Capítulo 3 describe el desarrollo de la aplicación *softphone XoIP* con base en los flujos de trabajo del UP.

En el Capítulo 4 se presentan los resultados obtenidos en el presente trabajo de tesis.

Las conclusiones y líneas futuras de investigación se presentan en el Capítulo 5.

Se presentan las referencias bibliográficas y los sitios de Internet consultados; finalmente se presenta el manual del usuario del *softphone XoIP* como anexo.. En el CD adjunto se incluyen los documentos complementarios al modelado de la herramienta software desarrollada y una descripción general de Asterisk.

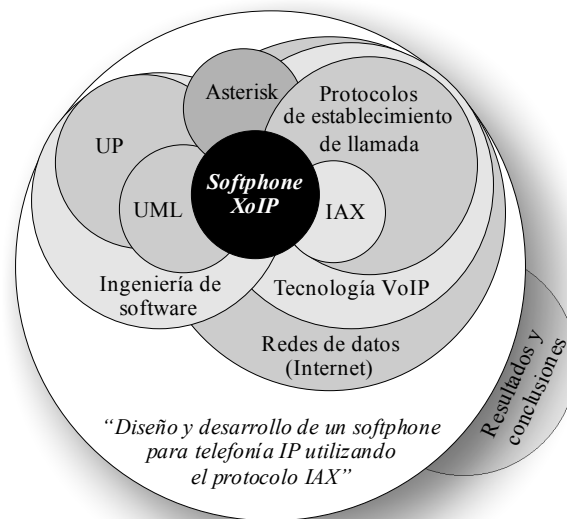


Figura 1.2. Estructura del documento de tesis.

1. Estado del arte de las comunicaciones de voz sobre IP

La voz sobre el protocolo de Internet, voz sobre IP o simplemente VoIP es una tecnología que permite conversaciones de voz mediante una conexión de Internet utilizando la línea telefónica. VoIP realiza la transmisión de voz a través del intercambio de paquetes de datos, para ello se definen los siguientes términos:

- *Voz sobre Internet*: Se refiere al establecimiento de llamadas sobre redes públicas (Internet) o redes privadas.
- *Telefonía IP*: Uso de IP para transmitir voz mediante el envío de paquetes de datos por Internet a la red pública de telefonía conmutada (PSTN, *Public Switched Telephone Network*). Es una aplicación inmediata de VoIP, la cual tiene como función la realización de llamadas telefónicas ordinarias sobre redes IP. “Con la telefonía IP la computadora se convierte en un teléfono” [URL15].
- *Paquetes de voz*: Uso de redes de datos en lugar de una red telefónica basada en conmutación de circuitos, para transportar llamadas telefónicas. La transmisión de voz mediante paquetes es un término general que puede aplicarse a cualquier tecnología de redes de datos, por ejemplo en Frame Relay (VoFR, *Voice over Frame Relay*) o en ATM (VoATM, *Voice over ATM*).

A pesar de que en el campo de las telecomunicaciones VoIP se conoce desde hace más de una década, es una tecnología que empieza a darse a conocer entre los usuarios de Internet y las micro, pequeñas, medianas y grandes empresas. Esta tecnología permite enviar información de voz en forma digital, en paquetes discretos haciendo uso del protocolo de Internet (IP, *Internet Protocol*).

Anteriormente, la transmisión de voz y la transmisión de datos eran dos conceptos teóricamente diferentes, pero con la revolución del Internet y los avances tecnológicos en redes de datos se ha podido establecer que dichos conceptos pueden convivir sobre una misma red, lo cual representa un gran ahorro para los hogares, instituciones educativas, oficinas y sucursales, proveedores y clientes dentro de un entorno empresarial, que cuentan con una red de datos.

La principal ventaja de VoIP es la transmisión de voz de forma gratuita, ya que ésta se transporta como si fueran datos, por lo tanto al poder transmitir voz sobre la red de datos hace que las empresas y los proveedores de servicios de internet (ISP, *Internet Services Provider*) puedan ofrecer valor agregado a sus clientes y revolucionar el comercio electrónico (*E-Commerce*). Con la tecnología VoIP cualquier empresa puede estar en contacto directo con sus clientes, proveedores, etc., simplemente necesitan agregar un acceso de comunicación mediante la página Web de la empresa y cualquier usuario puede comunicarse mediante el acceso de comunicación telefónica en línea; con este medio, un representante de la empresa puede brindar información en forma directa, haciendo más confiables las ventas por Internet.

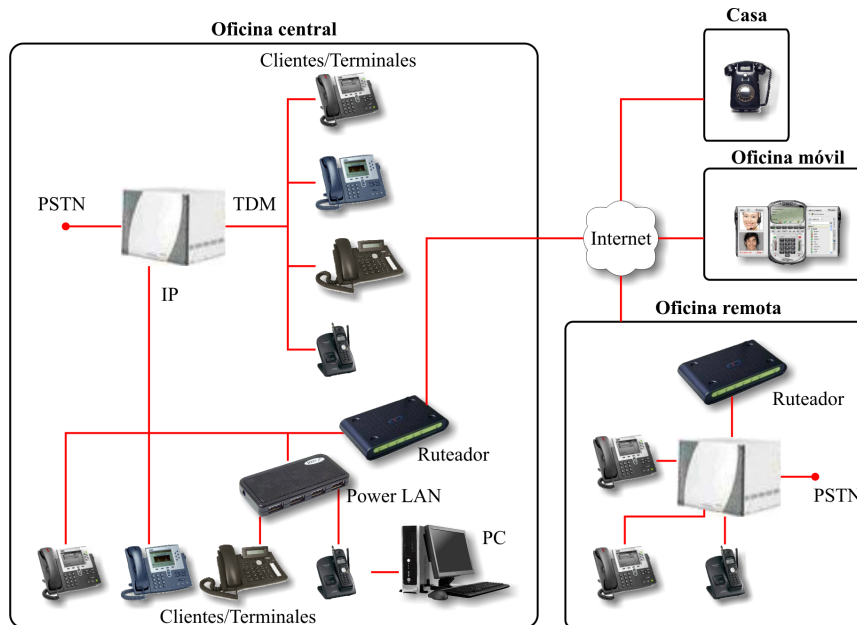


Figura 1.1. Comunicación VoIP entre sucursales.

Actualmente, las empresas utilizan VoIP para comunicaciones internas al ofrecer:

- *Acceso a redes corporativas desde pequeñas sedes a través de redes integradas de voz y datos:* Las empresas están totalmente comunicadas y las llamadas telefónicas entre sucursales a través de VoIP son gratuitas (Figura 1.1).
- *Agenda o directorio telefónico de la empresa y sucursales:* Herramienta que permite el establecimiento de una llamada sin necesidad de marcar el número de teléfono, ya que éste se encuentra almacenado en la memoria del teléfono (*softphone*).
- *Servicio de mensajes de voz y asignación de números personales:* Las personas de una empresa no necesitan estar dentro de las instalaciones para acceder a su número personal y revisar sus mensajes de voz, ya que VoIP brinda la posibilidad de realizar llamadas telefónicas a cualquier extensión de la empresa o a otro número de la PSTN, siempre y cuando se tenga un proveedor de servicios VoIP o un contrato con la compañía telefónica local (Telco, *Telephone Company*); sólo se requiere de una conexión a Internet y el *softphone* o *Hardphone* para marcar el número de sus contactos
- *Servicio de videoconferencias:* Los líderes o jefes ya no tendrán que trasladarse a algún lugar en especial para reunirse y hablar de los temas de gran relevancia para la empresa, sólo basta con entrar a una sala de videoconferencias a través de un *softphone* o teléfono IP para contactar al número de personas deseadas e iniciar la conversación entre ellas (Figura 1.2), ahorrándose costos de traslado, hospedaje y alimentación.



Figura 1.2. Videoconferencia.

1.1. Historia de la tecnología VoIP

La tecnología VoIP surgió en el año 1995, cuando la empresa VocalTec desarrolló el primer *softphone* para ejecutarse en una PC⁸; esta aplicación fue llamada Internet Phone e implementaba el protocolo de establecimiento de llamada H.323 [5].

A pesar del éxito obtenido, Internet Phone presentaba diversas desventajas ya que Internet no contaba con suficiente ancho de banda y la utilización de módems ocasionaba una calidad muy pobre en las conversaciones de voz comparado con una llamada telefónica a través de una línea de teléfono tradicional o PSTN; para establecer una comunicación de voz era necesario que ambos usuarios contaran con dicha aplicación debido a que ésta no soportaba herramientas de comunicación diferentes.

A finales de 1996, el uso de VoIP se consideraba una especie de entretenimiento en Internet. Aún no se establecían normas de comunicación, por lo que para realizar una conversación de voz mediante VoIP era necesario llamar primero por teléfono tradicional para averiguar si el otro usuario estaba conectado a Internet, por otro lado, en el mercado no existían herramientas que soportaran dicha tecnología.

En 1997, se desarrolló una pasarela (*gateway*) VoIP, la cual establece un puente de comunicación entre usuarios VoIP y usuarios de la PSTN, y permite realizar conversaciones de voz a cualquier teléfono, fijo o móvil.

En 1998 se estableció la utilización de la pasarela VoIP y se desarrolló el *gatekeeper*, logrando incrementar en 1% el tráfico de VoIP en los Estados Unidos de América. Se desarrollaron dispositivos y sistemas VoIP para realizar conversaciones de voz entre una PC y un teléfono tradicional, incluso para comunicar teléfonos tradicionales mediante una red IP.

En 2000, las firmas Cisco Systems y Lucent Technologies, fabricantes de sistemas de redes de comunicaciones de datos, lanzaron al mercado equipos que podían transportar paquetes de datos a una dirección destino dentro de Internet e intercambiaban tráfico de VoIP, lo cual incrementó a un 3% el tráfico de voz sobre una red de datos [URL19].

A partir de entonces se ha incrementado el número de fabricantes, proveedores y usuarios de tecnología VoIP. En el año 2005 se estimó un aumento en el uso de tecnología VoIP entre el 25% y el 40%. La mayoría de empresas están adaptando gran parte de sus redes al uso de tecnología VoIP con la finalidad de ofrecer un valor agregado a sus clientes, facilitar el uso de sus servicios y disminuir sus costos de comunicación [URL18].

Inicialmente el protocolo de establecimiento de llamadas era H.323, en la actualidad han surgido otros como SIP (*Session Initiation Protocol*), IAX (*Inter-Asterisk eXchange Protocol*), MGCP (*Media Gateway Control Protocol*), Megaco (*Media Gateway Controller*), etc.

SIP es el protocolo *de facto* debido a su calidad de comunicación, gran cantidad de dispositivos en el mercado y bajo costo de implementación. Se tiene contemplado que a corto plazo IAX sea el protocolo más utilizado, ya que cuenta con el respaldo de los desarrolladores de Asterisk⁹ [URL6] y considera diversas características que no soporta SIP.

⁸ PC basada en un microprocesador Intel 486 a 33 MHz, con una tarjeta controladora de audio, micrófono y bocinas.

⁹ Software libre que cumple con todas las funciones de una central de llamadas (*call center*).

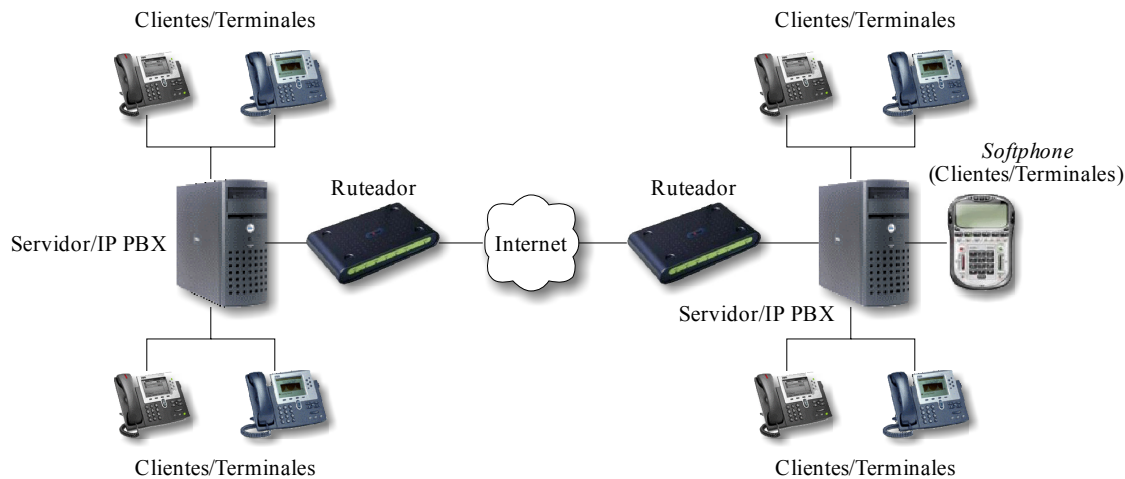


Figura 1.3. Elementos principales en la transmisión de VoIP.

1.2. Elementos de la tecnología VoIP

El modelo de VoIP involucra tres elementos principales [6]:

- *Cliente*: Se encarga de establecer y terminar las llamadas de voz; codificar, empaquetar y transmitir la información de salida que se genera por el micrófono del usuario; recibir, decodificar y reproducir la información de voz de entrada a través de las bocinas o audífonos del usuario. El teléfono puede presentarse como: *softphone*, el cual cuenta con todos los requerimientos que se necesitan para establecer llamadas de voz; o *hardphone*, que cumple con las normas de establecimiento de llamadas IP.
- *Servidor/IP PBX*: Encargado de controlar un gran número de operaciones de bases de datos en tiempo en real, las cuales incluyen validación de usuarios, distribución de utilidades, enrutamiento, administración general del servicio, carga de clientes, control del servicio, registro de usuarios y servicio de directorio. Este elemento es de vital importancia ya que sin él no se puede realizar la administración del registro y el establecimiento de las llamadas VoIP.
- *Pasarela*: Proporciona un puente de comunicación entre usuarios VoIP y usuarios PSTN. La pasarela origen debe transformar las señales de voz a datos digitales, dividirlos en paquetes para enviarlos a través del protocolo IP a la pasarela destino, la cual realiza el proceso inverso. La pasarela basa su funcionamiento en un procesador digital de señales (DSP, *Digital Signal Processor*) [5].

La pasarela establece la diferencia entre los conceptos telefonía IP y VoIP, ya que cuando la red de datos cuenta sólo con servidores y clientes, se considera el término VoIP. La Figura 1.3 muestra una configuración típica de un sistema VoIP, en donde los Clientes/Terminales pueden ser teléfonos analógicos, *hardphones* o *softphones*, estos últimos se ejecutan en una PC, Palm, PC pocket, etc. Los teléfonos analógicos necesitan de un dispositivo para digitalizar la señal de voz. En el caso de contar con los tres elementos que conforman la tecnología de VoIP, se establece como un sistema de telefonía IP (Figura 1.4).

1.3. Características de la señal de voz sobre IP

El envío de señales de voz en forma de paquetes de datos a través de Internet es similar a la transferencia de datos a través del sistema MSN (Messenger) o al envío de correo electrónico, ya que los paquetes de datos pasan a través de diferentes nodos hasta llegar al nodo destino.

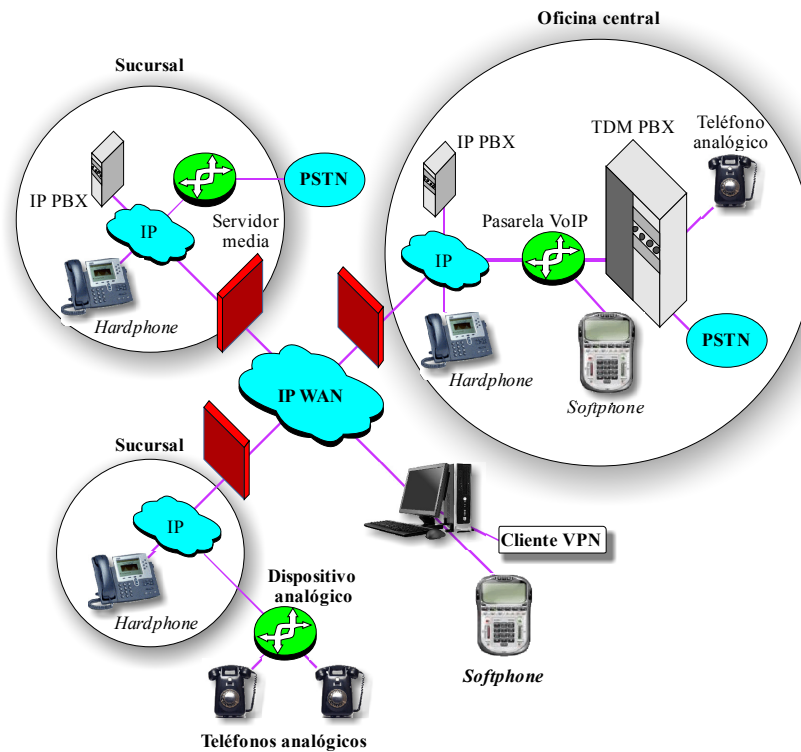


Figura 1.4. Configuración típica para la transmisión de voz a través de Telefonía IP.

En los ejemplos señalados, el envío de datos no es fiable y algunos paquetes pueden perderse durante el envío, sin embargo se permite al nodo destino solicitar la retransmisión del paquete correspondiente. En el caso de una conversación de voz no es válida la retransmisión de paquetes perdidos, ya que se requiere que los paquetes lleguen en el orden de envío para asegurar la coherencia de la conversación entre los participantes, y que la calidad de la voz recibida en ambos extremos sea audible, lo cual hace necesaria la implementación de sistemas QoS que evalúen la señal de voz.

1.3.1. Características de implementación de VoIP

Existen ciertas características que deben considerarse en la implementación de un sistema VoIP, las cuales se describen en los incisos siguientes.

1.3.1.1. Ancho de banda

La calidad de una conversación de voz depende del ancho de banda asignado. En los inicios de la tecnología VoIP, la transmisión de voz y video requería un elevado ancho de banda, lo cual hacía imposible establecer una comunicación de voz y video sobre la red de datos que garantizara una QoS aceptable.

Gracias a los avances tecnológicos en hardware y en software, actualmente se utilizan diversos algoritmos (GSM, G.711, G.723.1, G.729, etc.) para comprimir la señal de voz que recibe la pasarela, permitiendo el uso de la misma red de datos para el establecimiento de una conversación de voz entre uno o más participantes. Estos algoritmos se caracterizan por conseguir una buena compresión de datos disminuyendo el tiempo de latencia.

Por ejemplo, el algoritmo de compresión G.729 CS-ACELP consigue comprimir los paquetes de voz en 8 Kbps, los cuales se digitalizan y el protocolo IP agrega cabeceras para ser transportados a través de una red de datos, por lo que el ancho de banda se incrementa hasta 16 Kbps. Sin embargo, a diferencia de una conversación mediante la PSTN, se cuenta con la posibilidad de eliminar los silencios de la conversación.

La técnica conocida como supresión de silencios permite que el ancho de banda ocupado por la transmisión de paquetes se asigne sólo en los momentos en que los participantes están hablando y el resto del tiempo se libere, logrando con ello una reducción del ancho de banda promedio para la transmisión de un paquete de voz a 8 Kbps. Utilizando el algoritmo de compresión G.729 CS-ACELP en un ancho de banda de 64 Kbps se pueden establecer hasta 8 llamadas de voz en forma simultánea.

1.3.1.2. Calidad en la transmisión de voz

Para lograr una calidad aceptable en la transmisión de voz se deben considerar [5]:

- *Codecs*: Son algoritmos que se encargan de codificar y decodificar la voz.
- *Cancelación de eco*: Requisito indispensable para una conversación VoIP, se encarga de eliminar, de forma automática y en tiempo real, posibles ecos o ruidos que distorsionan la conversación de voz.
- *Latencia*: Tiempo necesario para que los paquetes viajen del nodo origen al nodo destino, incluyendo tiempos para comprimir, transmitir y descomprimir. Con la mejora que ha tenido la tecnología VoIP, la latencia es cada vez más pequeña llegando a los 120 ms.

1.3.1.3. Estándarización

Actualmente, las compañías de telecomunicaciones han establecido estándares destinados a la tecnología VoIP, con la finalidad de manejar y controlar el establecimiento de llamadas telefónicas a través de Internet. Tal es el caso de H.323, desarrollado por la ITU (*International Telecommunication Union*) y de SIP, desarrollado por la IETF (*Internet Engineering Task Force*).

El estándar H.323 fue desarrollado originalmente para comunicaciones multimedia (audio, video y datos) en una LAN, pero con el paso del tiempo ha migrado hacia redes más grandes. H.323 se encarga de controlar las llamadas y codificar la voz con base en las especificaciones que establece la transmisión de paquetes de voz en tiempo real.

En la actualidad el protocolo SIP presenta gran demanda debido a las siguientes ventajas: permite extrapolarse con otras aplicaciones, es un protocolo escalable, proporciona tiempos de respuesta mucho menores a los de H.323, su integración a Internet es prácticamente transparente y los equipos disponibles en el mercado son más baratos que los que implementan H.323.

Aunque SIP es el estándar más utilizado, existen estándares como IAX que presentan ventajas respecto a NAT y el uso de cortafuegos.

1.3.1.4. Regulaciones gubernamentales

Los servicios VoIP están inquietando a las grandes compañías telefónicas tradicionales debido al rápido crecimiento de los servicios de valor agregado que actualmente se ofrecen en Internet, como es el caso de la telefonía IP.

En México, la tecnología de VoIP está en pleno crecimiento y aunque la mayoría de empresas, instituciones, universidades, etc. la desconocen, ha despertado la preocupación de la empresa líder de telefonía en este país. En Europa, donde la tecnología VoIP está dando mayores frutos, la Unión Europea ha realizado diversas mesas de debates cuyos resultados han establecido que para que la telefonía IP pueda ser regulada debe cumplir con los siguientes criterios [5]:

- Las comunicaciones deben ser objeto de ofertas comerciales.
- Las comunicaciones deben ofrecerse al público en general.
- Las comunicaciones deben tener, como origen y destino, puntos de enlace con la PSTN.
- La comunicación implica transporte y conmutación en tiempo real.

La Unión Europea ha considerado que dichos criterios aún no se cumplen y por ello la telefonía IP no ha logrado su regulación. Sin embargo, cabe mencionar que existen proveedores de telefonía IP establecidos en países donde las leyes no les impiden proveer sus servicios como una compañía de teléfonos, por ejemplo Ubifone, Skype, net2phone, SitaTel, etc.

1.3.1.5. Principales aplicaciones de VoIP

VoIP incluye una serie de aplicaciones que actualmente tienen gran demanda y suponen un gran ahorro en costos de comunicación, las principales son:

- *Centros de llamadas por la Web*: Este tipo de aplicaciones está revolucionando el comercio electrónico. La mayoría de las empresas promocionan y venden sus productos y servicios a través de Internet; la principal desventaja es que los usuarios desconfían de la seguridad de las páginas Web al solicitar información confidencial de las tarjetas de crédito. Sin embargo mediante páginas seguras que presentan el icono *Push-to-Talk*¹⁰, el usuario puede navegar por la página y establecer comunicación directa con un agente del departamento de ventas, con lo que se logran aspectos de credibilidad, confianza y seguridad al realizar compras seguras. Las principales ventajas de este tipo de aplicaciones son: el establecimiento de la llamada no representa costo alguno al usuario, sólo se requiere una conexión a Internet y el cliente tiene comunicación directa con el personal de la empresa.
- *Redes privadas virtuales de voz (Voice-VPN)*: Este tipo de aplicación representa el menor costo de llamadas telefónicas, por ejemplo las entidades bancarias deben mantener comunicada toda la red de sucursales con su oficina central, y para ello es necesario interconectar las centralitas telefónicas de cada sucursal mediante la red IP corporativa, dando como resultado que todas las llamadas realizadas sean gratuitas debido a que hacen uso de la infraestructura establecida a través de Internet.
- *Fax*: Para la recepción de fax sobre redes de telefonía IP no es necesario contar con equipos específicos de la tecnología VoIP, es suficiente el empleo de un dispositivo de fax convencional. Para la transmisión es necesario contar con un equipo que soporte fax sobre IP (FoIP, *Fax over Internet Protocol*), el usuario se encarga de colocar el documento a enviar, indicar la lista de números telefónicos destino, y el sistema se encarga de digitalizar, enviar y enrutar los paquetes de datos a los nodos destino.
- *Multiconferencia*: Hace referencia al establecimiento de una conversación de voz y datos entre tres o más personas simultáneamente, en donde los participantes pueden revisar documentos, charlar y dar sus puntos de vista al respecto. Se considera como un valor agregado de la telefonía IP, en el cual las personas no tienen que abandonar sus oficinas para asistir a reuniones realizadas fuera de su lugar de trabajo.

1.3.2. Ventajas y desventajas sobre el uso de VoIP

La tecnología VoIP presenta las siguientes ventajas:

- *Proporciona un número de teléfono único*: Con este número el usuario puede comunicarse desde cualquier lugar donde haya una conexión a Internet.
- *Ahorro en llamadas de larga distancia*: Una llamada de PC a PC o entre sucursales conectadas mediante la red IP corporativa no representa costo alguno, el único costo que se incluye es el de los servicios de Internet.

¹⁰ La tecnología presionar para hablar (PTT o P2T, *Push-to-Talk*) es un método para establecer una comunicación de voz en líneas *half-duplex*, oprimiendo un botón para establecer una llamada a través de la Web.

- *Llamadas a teléfonos fijos y celulares*: La comunicación de VoIP no se limita a llamadas de PC a PC con conexión a Internet, sino que permite comunicarse a teléfonos de cualquier parte del mundo, ya sean fijos o celulares, con un ahorro aproximado del 80% sobre el costo de la llamada.
- *Mensajería unificada y correos de voz*: En situaciones en que no se pueda hacer uso de un teléfono, se pueden consultar los mensajes de voz mediante una conexión a Internet y si es necesario se puede realizar la llamada al número telefónico del mensaje correspondiente mediante del *softphone*.
- *Variedad de aplicaciones*: Las cuales permiten estar comunicado en todo momento, por ejemplo fax y/o multiconferencia.
- *Marketing y servicios al cliente*: Mediante la filosofía PTT, los clientes tienen comunicación telefónica directa con la empresa al mismo tiempo que navegan y exploran los productos o servicios ofrecidos en la página Web.
- *Trabajo a distancia (teletrabajo)*: Es un nuevo concepto destinado a las bolsas de trabajo, en el que las personas pueden realizar su trabajo desde casa mediante una conexión IP con la red de la empresa.
- *Servicios a usuarios domésticos*: Estos usuarios pueden utilizar la línea telefónica, navegar por Internet, identificar y desviar llamadas, etc.
- *Servicios XoIP*: Hace referencia a un nuevo conjunto de servicios sobre IP, donde la “X” puede ser voz, video, datos, fax, etc.

La tecnología VoIP presenta las siguientes desventajas:

- *QoS*: Es la principal desventaja que presenta la tecnología VoIP y depende de los siguientes factores [5]:
 - *Retardo o latencia*: Tiempo necesario para comprimir, transmitir y descomprimir los paquetes de voz; debe tener un rango de 120 a 150 ms para que la conversación tenga coherencia.
 - *Eco*: Por lo regular se debe al mal acoplamiento de impedancias.
 - *Supresión de silencios*: La actividad de detección de voz (VAD, *Voice Activity Detection*) es un factor importante encargado de eliminar el envío de paquetes de silencio sobre la red IP y de detectar sólo los tiempos de conversación para que puedan ser enviados como paquetes de datos, con lo cual se libera ancho de banda.
 - *Pérdida de paquetes de voz*: Este fenómeno provoca pérdida de coherencia en la conversación ya que el destino no cuenta con todos los paquetes de voz enviados desde el origen.
 - *Priorización de paquetes en la red IP*: En una red IP interna se pueden emplear técnicas de priorización de paquetes como son: CQ (*Custom Queuing*), asigna un porcentaje del ancho de banda disponible; PQ (*Priority Queuing*), establece prioridad en las colas; WFQ (*Weight Fair Queuing*), asigna prioridad al tráfico de menos carga; y DiffServ¹¹, evita tablas de ruteadores intermedios y establece decisiones de rutas por paquetes.

¹¹ DiffServ, o diferenciación de servicios, es una arquitectura que especifica un método para proporcionar QoS, garantizando el tráfico de una red IP. Puede ser usado para proporcionar baja latencia y garantizar el servicio transferencia de voz y video en una red con tráfico crítico, mientras proporciona su mejor esfuerzo garantizando el tráfico de servicios no críticos como es el tráfico Web o la transferencia de archivos.

1.4. Razones que fundamentan el uso de VoIP

La telefonía por Internet se presenta como una tecnología atractiva y novedosa dentro del mercado de las comunicaciones, sin embargo en algunos sectores se desconfía de la calidad de las conversaciones de voz, por lo que existen esfuerzos en atender los aspectos técnicos y consolidar dicha tecnología como una solución viable a las demandas del mercado. Con base en lo anterior, los esfuerzos se enfocan en cuatro factores que fundamentan la implantación de VoIP, el caso comercial, la presencia universal de IP, la consolidación de las tecnologías empleadas y el cambio a las redes de datos, los cuales se describen en los apartados siguientes.

1.4.1. Caso comercial

Existen cuatro requerimientos para implantar el conjunto de protocolos VoIP y el equipo de soporte a los servicios de telefonía [3]:

- *Integración de voz, datos y video*: La mayoría de aplicaciones software demandan la integración de voz, datos y video, revolucionando el diseño y desarrollo de los servidores Web con la finalidad de interactuar con el cliente y ofrecer nuevas soluciones en tiempo real.
- *Asignación del ancho de banda*: La integración de voz y datos permite estandarizar el uso del ancho de banda con la finalidad de mejorar la utilización de los canales de comunicación. La idea principal es no utilizar la telefonía convencional, basada en el esquema de multiplexación de división en el tiempo (TDM, *Time Division Multiplexing*) en donde un usuario tiene asignado un ancho de banda en forma interrumpida, incluso cuando no está hablando. Considerando que una conversación de voz implica aproximadamente un 50% de tiempo de silencios, se emplea un esquema de TDM estático (STDM, *Statistical Time Division Multiplexing*), el cual utiliza el ancho de banda cuando es necesario y se libera instantáneamente para ser utilizado por otro usuario en los tiempos de silencios. Además se considera que en la conversación existe otro 20% de patrones repetitivos, los cuales se pueden eliminar mediante algoritmos de compactación [3]. Existen otras técnicas características de las redes basadas en paquetes que se implementan en la asignación del ancho de banda, optimizando su uso en una escala de 10:1.
- *Capacidad de negociación y adaptación*: La infraestructura actual de la telefonía está basada en una tecnología desarrollada hace décadas, la cual es rígida, estática y difícil de adaptarse a los cambios y servicios emergentes. La tecnología VoIP presenta una infraestructura flexible para soportar diferentes tecnologías y la habilidad de negociar servicios. Por ejemplo, los usuarios pueden negociar el uso de diferentes tarifas de datos, tecnologías de codificación o cifrado de datos, direcciones IP, números de puerto, requerimientos de retardos, incrementar operaciones de calidad de servicio, etc.
- *Tarifas arbitrarias*: Se refiere a considerar las tarifas de Internet en lugar de las establecidas por el uso de la PSTN. Los desarrolladores de VoIP esperan que la Comisión Federal de Comunicaciones (FCC, *Federal Communications Commission*) elimine las comisiones por proporcionar servicios mejorados (ESP, *Enhanced Service Providers*) a los ISPs con la finalidad de reducir sus costos de operación y lograr una competencia justa respecto a los servicios de VoIP.

1.4.2. Presencia universal de IP

A diferencia de las tecnologías ATM y Frame Relay que operan como interfaces de usuario a red (UNI, *User Network Interface*), la familia de protocolos TCP/IP se ha consolidado a nivel global y está presente en la gran mayoría de lugares de trabajo (Figura 1.5) [20, 21].

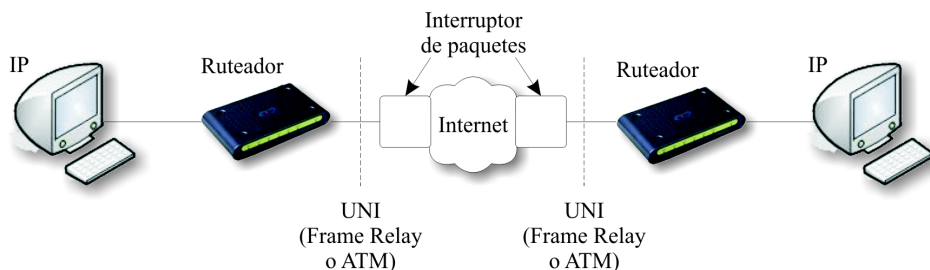


Figura 1.5. Localización de IP vs ATM y Frame Relay¹².



Figura 1.6. Descripción de ATM en la red.

La tecnología VoIP opera tanto en LANs como en WANs, mientras que Frame Relay y ATM operan sólo en WANs, sin embargo se debe considerar que los proveedores de servicios de telefonía implementan ATM en muchas de sus redes troncales (*backbones*). Las redes troncales comprenden todos los componentes de red que operan entre los nodos de la compañía de teléfonos, a excepción de los componentes que operan en el lazo local (*local loop*) del cliente, y se extienden desde las oficinas centrales telefónicas (Figura 1.6) [3].

Los proveedores de servicio de telefonía utilizan ATM debido a que proporciona varias herramientas para administrar el tráfico en una red y facilita la operación correcta con la tecnología convencional de telefonía.

1.4.3. Maduración de las tecnologías

Otro factor importante es la maduración de las tecnologías que hacen factible la telefonía IP, sobre todo por el avance en gran escala de los DSP que forman parte de codecs y módems de alta velocidad, cuyo funcionamiento y velocidad de operación permiten la implementación de una gran cantidad de aplicaciones de usuario.

Actualmente, el usuario final demanda el uso de aplicaciones que soportan imágenes en tres dimensiones, operaciones en tiempo real, video de movimiento completo (*full-motion*) y despliegue de datos. Asimismo, se consolidan tres tecnologías claves: incremento de la capacidad de las computadoras, aumento de las capacidades de acceso a la comunicación y reutilización de componentes en el desarrollo de código software; la convergencia de dichas tecnologías ha iniciado una nueva generación de aplicaciones de usuario.

1.4.4. Migración a redes de datos

Otra razón del éxito de la tecnología VoIP es el hecho de que el mundo está experimentando una migración de redes basadas en circuitos (red de telefonía) a redes basadas en paquetes (redes de datos), debido principalmente a dos razones: la industria de datos presenta un mayor crecimiento respecto a la industria de la voz¹³ debido a que en la actualidad la gente tiene mayor acceso a las computadoras que a los teléfonos; y los requerimientos de ancho de banda y QoS para una conver-

¹² El término conmutador de paquetes (*packet-switch*) se emplea en forma genérica, pueden emplearse los términos conmutador Frame Relay (*Frame Relay switch*) o conmutador ATM (*ATM switch*).

¹³ Se considera a la red de telefonía como la industria de la voz.

sación de voz no se ven afectados mientras esté funcionando la telefonía por circuitos entre dos usuarios.

Por otra parte, se requiere mayor ancho de banda y QoS para conversaciones que se realizan desde la PC, por lo que es necesario expandir las redes de datos para satisfacer dichas necesidades y contar con computadoras adecuadas para soportar el uso de tecnología VoIP.

1.5. Codecs de voz

Los codecs de voz son modelos matemáticos usados digitalmente para codificar y comprimir la información de audio analógica. Muchos de estos modelos matemáticos se basan en la capacidad del cerebro humano de obtener la información a pesar de que ésta sea incompleta. Comúnmente conocidos como codecs, *speech coders* o *voice coders*, son parte fundamental en el manejo y uso de la tecnología VoIP, y se encargan de codificar/decodificar la voz [22].

1.5.1. Funciones de los codecs

La función principal de un codec es codificar las muestras de la conversación del usuario en tramas mediante un código de modulación de pulso (PCM, *Pulse Code Modulation*), de manera que la conversación logre las ventajas que presenta un sistema digital, como pueden ser: mejorar la señal en presencia de errores en la comunicación y disminuir la inestabilidad en las redes y transmisiones ruidosas. En el receptor, las tramas son decodificadas para obtener las muestras de conversación de PCM y después convertirlas a formas de onda (señal de voz).

1.5.2. Clasificación de los codecs

Los codecs se clasifican en tres tipos:

- *Codec en forma de onda*: Intenta reproducir fielmente la señal de voz, incluyendo los ruidos de fondo, mediante el aprovechamiento de sus características temporales o espectrales, con la finalidad de realizar la codificación en forma eficiente; asimismo presenta alta calidad y un ancho de banda en el rango de 16-64 Kbps. Por ejemplo: PCM, DPCM (*Differential Pulse Code Modulation*), ADPCM (*Adaptive Differential Pulse Code Modulation*), codificador en sub-bandas, etc.
- *Vocoder (voice + coders, voz + decodificador)*: No reproduce la forma de la señal (*waveform*) original. El codificador construye una serie de parámetros que son enviados al receptor para utilizarse como manejadores de un modelo de producción de conversación. Se emplea una codificación de predicción lineal (LPC, *Linear Prediction Coding*) para deducir los parámetros de un filtro digital en diferentes tiempos. La calidad del vocoder no es suficientemente buena para implementarse en los sistemas de telefonía, sin embargo asume un modelo para la señal de voz; obtienen y cuantifican los parámetros de dicho modelo. Presenta una velocidad de bit (*bit rate*) en el rango de 1.2-4.8 Kbps y baja calidad. Ejemplos de este tipo de codecs son: Vocoder LPC y Vocoder homomófico [URL8].
- *Codec híbrido*: También conocido como codificador de análisis por síntesis (AbS, *Analysis by Synthesis*), es el tipo de codificador más empleado en la actualidad, ya que combina las características de los codificadores de forma de onda con las del vocoder con la finalidad de obtener una alta calidad de voz a bajas velocidades de bit (4-16 Kbps). En estos codecs las muestras de la señal de entrada se dividen en bloques de muestras (vectores), los cuales son procesados como si fueran uno solo; lleva a cabo una representación paramétrica de la señal de voz para intentar que la señal sintética sea lo más parecida a la original. Ejemplos de este tipo de codecs son: CELP (FS-1016), Multipulso (MPC), VSELP, RELP, RPE-LTP, etc. [3].

1.5.3. Evaluación de codecs

Se deben considerar los siguientes factores para evaluar el funcionamiento y rendimiento de los codecs:

- *Tamaño y retardo de la trama*: Representa la longitud del tráfico de voz medido en tiempo. Las tramas son partes discretas de la conversación y cada una se actualiza de acuerdo con las muestras de la conversación, posteriormente se insertan en paquetes que son enviados al receptor.
- *Retardo de procesamiento o algoritmo de retardo*: Representa el retardo que introduce el codec en la ejecución de sus tareas.
- *Retardo de anticipación (Look ahead)*: La idea del retardo de anticipación es tomar ventaja sobre las correlaciones existentes entre tramas de voz sucesivas y ocurre cuando el codec examina cierta cantidad de tramas subsiguientes para proporcionarles dirección durante la codificación de la trama actual.
- *Longitud de la trama*: Este valor representa el número de octetos (*bytes*) resultantes del proceso de codificación, sin considerar las cabeceras de la trama.
- *Velocidad de bit*: Este parámetro es el índice de salida del codec cuando en su entrada está presente una señal PCM de voz estándar.
- *DSP MIPS (Million Instruction Per Second)*: Este valor define la velocidad mínima para que el DSP pueda soportar un codificador específico. Los DSP MIPS están diseñados para trabajar en un codec específico.
- *RAM requerida*: Este valor describe la cantidad de RAM necesaria para soportar un proceso de codificación específico.

1.5.4. Comparativa de codecs

La elección del codec a utilizar es un tanto subjetiva por parte de los usuarios o desarrolladores, y puede basarse principalmente en la utilización del ancho banda y en la velocidad de operación. A continuación se describen las soluciones más utilizadas [URL17].

1.5.4.1. ILBC

El ILBC (*Internet Low Bitrate Codec*) es gratuito y apropiado para comunicaciones de voz robustas sobre IP, está diseñado para hacer uso de un ancho de banda reducido dependiente del tamaño de muestra utilizada (20 o 30 ms); por ejemplo, si trabaja con tramas de 20 ms requiere de un ancho de banda de 15.20 Kbps (303 bits empaquetados en 38 octetos) y si trabaja con tramas de 30 ms reduce el ancho de banda a 13.33 Kbps (399 bits empaquetados en 50 octetos). Respecto a calidad, permite una degradación suave de la voz ocasionada por el retraso o la pérdida de paquetes, dicha degradación se logra mediante la extra/interpolación de los paquetes faltantes.

Existen pocos dispositivos comerciales¹⁴ ya que es una solución nueva, y debido a su complejidad y reducido ancho de banda, requiere de una cantidad importante de procesamiento, lo que impide mantener varias llamadas al mismo tiempo [URL9].

1.5.4.2. g.729a

El codec G.729 emplea 8 Kbps como ancho de banda y a diferencia del ILBC, G.729 es un codec propietario por lo que su implementación y uso requiere de una licencia; a pesar de ello es soportado por una gran variedad de dispositivos comerciales.

¹⁴ GrandStream es una de las principales firmas que brinda soporte a nivel comercial.

Se considera el codec estándar en aplicaciones de bajo consumo de ancho de banda. Utiliza una técnica conocida como CS-ACELP (*Conjugate Structure-Algebraic Code Excited Linear Prediction*), la cual reduce el tamaño de la señal de entrada en una razón de 8:1 (8 Kbps en lugar de los 64 Kbps de un canal de voz). Su calidad es parecida a la proporcionada por el sistema global para comunicaciones móviles (GSM, *Global System for Mobile Communication*). Cumple con los requisitos que establece una conversación telefónica, aunque no ofrece un buen soporte respecto a la degradación de la voz ocasionada por el retraso o la pérdida de paquetes.

1.5.4.3. g.711 (*u-law* y *a-law*)

Este codec ofrece la máxima calidad de conversación ya que requiere un ancho de banda de 64 Kbps, un canal de voz en la red digital de servicios integrados de banda ancha (RDSI-BA)¹⁵ y no utiliza técnicas de compresión. El hecho de no emplear técnicas de compresión evita requerimientos de procesamiento y reduce los costos de implementación. Su principal aplicación es la comunicación entre nodos de una misma LAN, ya que facilita el proceso de establecimiento de llamadas y la utilización del ancho de banda es constante. G.711 está estandarizado por la ITU y en consecuencia existen varias soluciones comerciales.

1.5.4.4. GSM

GSM es un sistema de comunicaciones celulares y posee su propio codec, el cual se conoce como excitación por pulsos regulares con predicción a largo plazo (RPE-LTP, *Regular Pulse Excitation Long-Term Prediction*). El codec GSM utiliza información de muestras previas para predecir la muestra actual; la señal de voz se divide en bloques de 20 ms, los cuales se envían al codec para su compresión. Los paquetes de voz son bloques de 33 octetos que con los intervalos de muestreo se obtiene un ancho de banda de 13.3 Kbps.

1.5.4.5. Speex

Speex es un codec gratuito para la compresión de audio; inició como proyecto de código abierto con la finalidad de adoptar la tecnología VoIP y ofrecer una solución gratuita ante los codecs propietarios existentes en el mercado. Speex se basa en CELP (*Code Excited Linear Prediction*) y está diseñado para comprimir la voz con un rango de ancho de banda de 2 a 44 Kbps; utiliza diferentes anchos de banda: banda angosta (8 Kbps), banda ancha (16 Kbps) y banda ultra-ancha (32 Kbps); además soporta codificación estereo, manejo de paquetes perdidos, operaciones de ancho de banda variable (VBR, *Variable Bitrate Operations*), VAD, transmisión discontinua (DTX, *Discontinuous Transmission*), puerto de punto fijo en curso, etc.

La Tabla 1.1 muestra una comparativa y la Tabla 1.2 muestra el consumo de ancho de banda de diferentes codecs VoIP recomendados por la ITU-T [3, URL1, URL5, URL9, URL14, URL16, URL17].

¹⁵ ISDN-B (*Integrated Services Digital Network-Broadband*) es un estándar para la transmisión de voz, video y datos a través de la línea telefónica de fibra óptica, y puede soportar velocidades de transferencia de datos de hasta 1.5 Mbps.

Tabla 1.1. Comparativa de codecs.

Codec	Frecuencia de muestreo (KHz)	BW (Kbps)	BW nominal (Kbps)	Tamaño del bloque (ms)	Licencia	Observaciones	Ventajas	Desventajas
DV14	Variable	32	N/A	Muestreada		No muy común, utiliza ADPCM		
G.711	8	64	87.2	Muestreada	Open source	G.711u/a comúnmente referidos como u-law/a-law: donde a-law es la versión europea y u-law es la versión de US/Japón	Diseñado para entregar máxima calidad de voz; muy bajo consumo de CPU	Incluyendo sobrecarga consume más de 64 Kbps, por lo tanto, requiere 128 kbps de ancho de banda en cada dirección (subida y bajada)
G.722	16	48 56 64	N/A	Muestreada	Open source	Un codec estándar de la ITU	Puede ser integrado en un circuito integrado y su retraso total es alrededor de 3ms, suficientemente pequeño para no causar eco en las redes de telecomunicaciones	Requiere mucho ancho de banda
G.723.1	8	5.3 6.3	20.8 21.9	30	Propietario	Comúnmente usado por proveedores de VoIP	Compresión muy alta, manteniendo excelente calidad de audio	Consume demasiado tiempo de procesamiento
G.726	8	16 24 32 40	N/A 47.2 55.2 N/A	Muestreada	Open source	Versión mejorada de G.721 y G.723 (totalmente diferente de G.723.1)	Se utiliza en troncales internacionales para ahorrar ancho de banda	Los gastos de procesamiento son relativamente bajos para el nivel de compresión obtenido
G.728	8	16	31.5	2.5	Open source	ITU	Tiene un algoritmo de retardo de codificación de 0.625 ms	Comparado con G.721 tiende a ser peor
G.729	8	8	31.2	10	Patentado	ITU	Excelente utilización del ancho de banda para conversaciones de calidad; funciona bien bajo errores de bits aleatorios	Requiere licencia para su uso
GSM	8	13	N/A	20	Propietario	Misma codificación usada en teléfonos celulares GSM	Alto porcentaje de compresión; se usa en hardware y software disponibles en el mercado de VoIP	La información que toma de muestras previas para predecir la actual, no cambia rápidamente.
iLBC	8 N/A	13.33 15.2	N/A N/A	30 20	Libre	Para uso de ancho de banda reducido	Alta robustez contra pérdida de paquetes	Requiere de mucho procesamiento lo que impide tener varias llamadas a la vez
Siren	N/A	N/A	N/A			No hay mucha información y no parece ser soportado		
Speex	8 16 32	2.15-24.6 4-44.2 N/A	30 34 N/A		Open source	Soporta codificación estereo, manejo de paquetes perdidos y detección de actividad de voz	Usa una velocidad de bit (<i>bitrate</i>) variable para minimizar el consumo de ancho de banda	Consume demasiado tiempo de procesamiento

Tabla 1.2. Soporte Cisco de codecs de la ITU-T.

Codec	BW* (Kbps)	MOS ⁺	Retar- do [%] (ms)	Tamaño del paquete ^{&} (octetos)	Carga útil [#] (octetos)	Paquetes por segundo [@]	Cabeceras IP/UDP/RTP [€] (octetos)	cRTP [¥] (octe- tos)	L_2 [§]	Cabecera L_2 [±] (octetos)	BW total No VAD ^µ (Kbps)	Total BW VAD [€] (Kbps)
G.729	8	3.9	15	10	20	50	40	2	Ether	14	29.6	14.8
							40				14.4	7.2
							40	2	PPP	6	26.4	13.2
							40				11.2	5.6
							40	2	FR	4	25.6	12.8
							40				10.4	5.2
40	2	ATM	2 celdas	42.4	21.2							
40			1 celda	21.2	10.6							
G.711	64	4.1	1.5	160	160	50	40	2	Ether	14	85.6	42.8
							40				70.4	35.2
							40	2	PPP	6	82.4	41.2
							40				67.2	33.6
							40	2	FR	4	81.6	40.8
							40				66.4	33.2
40	2	ATM	5 celdas	106.0	53.0							
40			4 celdas	84.8	42.4							
G.729	8	3.9	15	10	30	33	40	2	PPP	6	20.3	10.1
							40				10.1	5.1
							40	2	FR	4	19.7	9.9
							40				9.6	4.8
							40	2	ATM	2 celdas	28.3	14.1
							40			1 celda	14.1	7.1
G.723.1	6.3	3.9	37.5	30	30	26	40	2	PPP	6	16.0	8.0
							40				8.0	4.0
							40	2	FR	4	15.5	7.8
							40				7.6	3.8
							40	2	ATM	2 celdas	22.3	11.1
							40			1 celda	11.1	5.6
40	2	PPP	6	13.4	6.7							
40				6.7	3.4							
G.729	5.3	3.65	37.5	30	30	22	40	2	FR	4	13.1	6.5
							40				6.4	3.2
							40	2	ATM	2 celdas	18.7	9.4
							40			1 celda	9.4	4.7

* Voice Bandwidth (BW) Kbps: ancho de banda que consume el codec.

+ MOS (Mean Opinion Score): puntuación sobre la calidad del codec (varía entre 0 y 5, se recomienda mayor a 4).

% Codec delay (ms): retardo ocasionado por el codec.

& Packet size (octetos): tamaño del paquete en octetos.

Cisco payload (octetos): información redundante para identificar y administrar las muestras de voz (incluye los octetos del tamaño del paquete).

@ Packets per second: porcentaje de emisión del codec y del tamaño de la carga útil.

€ IP/UDP/RTP headers (octetos): número de octetos de las cabeceras IP, UDP y RTP.

¥ cRTP (octetos): información redundante de la utilización del algoritmo de compresión de Cisco.

§ L_2: tipo de servicio de portador de la capa 2.

± L_2 Headers (octetos): cabeceras de la capa 2.

µ Total Bandwidth (BW) (Kbps) no VAD: ancho de banda requerido con la opción no VAD [3].

€ Total Bandwidth (BW) (Kbps) with VAD: ancho de banda requerido con la opción VAD [3].

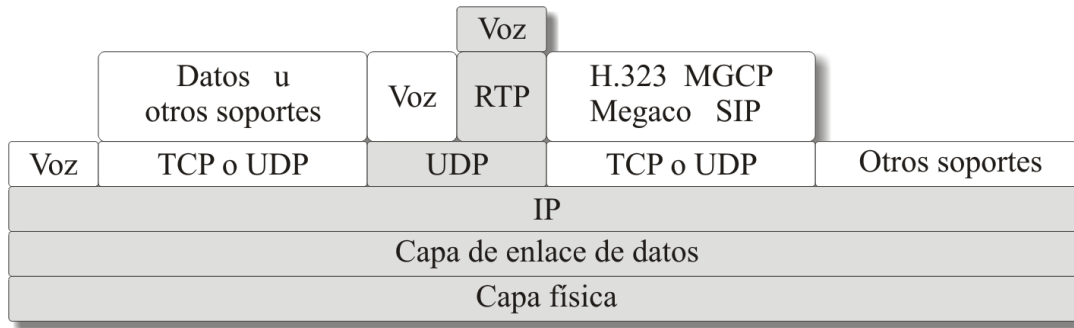


Figura 1.7. Conjunto de protocolos VoIP correspondientes a la familia de protocolos de Internet.

1.6. Modelo VoIP

La Figura 1.7 muestra el modelo de capas de la familia de protocolos de Internet relacionados con VoIP [3]. En el caso de la voz, ésta puede transportarse mediante tres caminos: sobre IP (Apartado 1.6.1), sobre UDP/IP (Apartado 1.6.2) o sobre RTP/UDP/IP (Apartado 1.6.3); desde el punto de vista técnico, el tráfico de voz puede funcionar sobre los tres caminos, sin embargo se recomienda el último. Por lo anterior, se considera que el término VoIP no es preciso debido a que no hace referencia al camino utilizado, y algunos proponen el término Voz sobre RTP (VoRTP, *Voice over Real Time Protocol*); sin embargo, y debido a la presencia universal del IP, se ha estandarizado el término VoIP.

1.6.1. Transmisión de voz sobre IP

Transmitir directamente la voz sobre IP significa encapsular el tráfico de voz en el campo del usuario correspondiente al datagrama IP [3].

1.6.2. Transmisión de voz sobre UDP

UDP es un protocolo importante para operaciones VoIP debido a que se encarga de administrar y controlar los números de puertos en Internet entre computadoras y aplicaciones, e identifica a estas últimas como la capa de aplicación que se ejecuta sobre el UDP.

La cabecera del datagrama UDP contiene los números de puerto, origen y destino, requeridos para la ejecución apropiada de los protocolos de la capa de aplicación. El número de puerto es concatenado con la dirección IP para formar el *socket*, dicha dirección tiene que ser única en Internet y un par de *sockets* identifica a cada punto final de la conexión (*end-point connection*).

Aunque el mapeo de puertos destinado a procesos de capas superiores puede ser manejado como asunto interno en un *host*, Internet publica los números de puertos reservados para procesos de más alto nivel de uso frecuente conocidos como puertos bien conocidos (*well-known ports*). Los *sockets* identifican las sesiones entre las aplicaciones finales.

Otro punto a considerar respecto al uso de UDP es que algunos protocolos de señalización basados en VoIP no pueden funcionar correctamente sin el uso de puertos. Por ejemplo, una de las funciones de SIP es soportar el paso de números de puertos entre aplicaciones, los cuales se utilizan durante la llamada telefónica de paquetes (*packet telephone call*). Finalmente, resulta apropiada la utilización de puertos y por ello, UDP es parte del camino de VoIP [3].

1.6.3. Transmisión de voz sobre RTP

RTP está diseñado para soportar tráfico de voz y video en tiempo real, brinda soporte a aplicaciones unidifusión (*unicast*) y multidifusión (*multicast*), proporciona servicios que incluyen la identificación del tipo de la carga útil (por ejemplo, el tipo de tráfico de audio como G.723 o G.729),

numeración de secuencia, sello de tiempo (*timestamping*) y monitoreo de entrega.

Existen aplicaciones que usualmente ejecutan RTP sobre UDP para realizar la suma de comprobación (*checksum*) y la multiplexación de puertos UDP. RTP soporta transferencia de datos para múltiples destinos usando multidifusión. Mediante los números de secuencia, el receptor puede reconstruir la secuencia de paquetes del remitente y conocer la localización de un paquete, además RTP proporciona funciones útiles a los usuarios de VoIP [3].

1.6.4. La transmisión de voz sobre TCP no es recomendable

Para aplicaciones de voz no se utiliza el protocolo de control de la transmisión (TCP, *Transmission Control Protocol*) debido a que éste provoca retardos en el tráfico. TCP soporta retransmisión y con ello controla temporizadores (*timers*) que no satisfacen los requisitos de tráfico en tiempo real. Por el contrario, UDP es un protocolo que no soporta retransmisión y por lo tanto no añade retardos que puedan afectar el tráfico de voz [3].

La notación “otros soportes” de la Figura 1.7 hace referencia a diversos protocolos que proporcionan funciones de ayuda para las operaciones de VoIP. La notación “datos” corresponde al hecho de que las sesiones VoIP soportan aplicaciones de datos, por ejemplo la transferencia de archivos, diapositivas, dibujos, hojas de cálculo, etc.

1.6.5. VoIP en Web

El modelo VoIP incluye al protocolos HTTP, lenguajes como HTML y XML, y el uso de URLs y DNS, los cuales son componentes esenciales (Figura 1.8).

La firma de telecomunicaciones CEO acuñó el término “Tono Web¹⁶” para describir el uso de tonos de marcado de la telefonía convencional en la Web. La tecnología VoIP soporta servicios Web y prevee la migración a redes de telefonía basada en tonos.

1.6.6. Protocolos VoIP agrupados en planes

La Figura 1.9 muestra al grupo de protocolos VoIP agrupados en tres planes, los cuales se basan en sus funciones e interfaces de los protocolos asociados [3]:

- *Plan de señalización*: Este plan contiene los protocolos de procesamiento o establecimiento de llamadas para VoIP, por ejemplo: Megaco, MGCP, H.323 y SIP.
- *Plan de soporte*: Contiene los protocolos VoIP que soportan la llamada y el plan de señalización, por ejemplo: RSVP, RTP y NTP.
- *Plan del usuario*: Contiene el tráfico generado por el usuario mediante paquetes de voz.

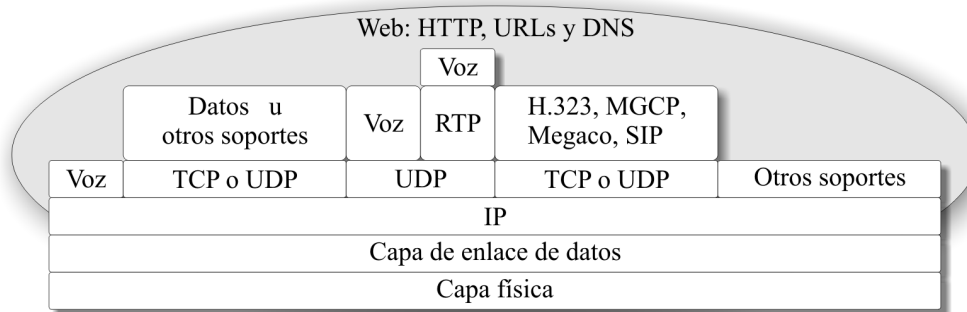


Figura 1.8. VoIP basada en Web.

¹⁶ Hace referencia a los servidores Web que utilizan el intercambio de telefonía.

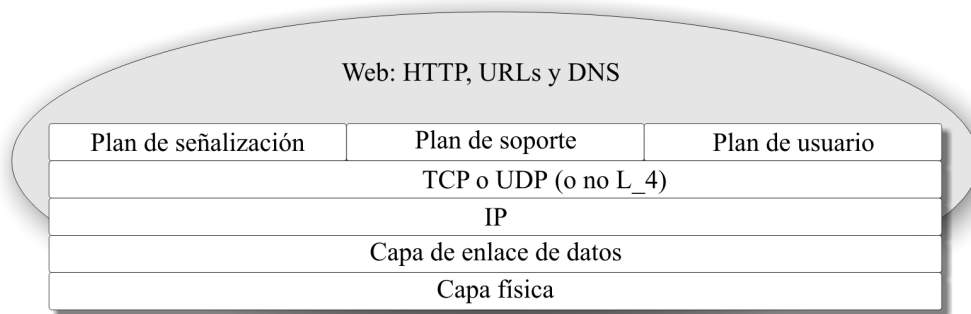


Figura 1.9. Modelo basado en Web agrupado en tres planes.

2. Protocolos de establecimiento de llamada

VoIP se basa en el proceso de dividir la señal de voz en paquetes (*packetization*) de audio *streams* para su transporte sobre Internet. Dichos paquetes deben llegar en el mismo orden en que fueron enviados y en un tiempo menor a 300 ms, ya que en caso de que algunos paquetes se pierdan o sufran retrasos, se degrada la calidad de la comunicación.

Los protocolos de transporte, conocidos colectivamente como Internet, fueron diseñados para prever la resolución de pérdida de paquetes mediante una solicitud, tiempo de espera, retransmisión o en algunos casos continuar con el procesamiento sin la información de los paquetes perdidos; las conversaciones de voz no permiten la pérdida de paquetes ni los retardos que surgen con la espera, por ello dichos protocolos no pueden brindar soporte a *streaming*¹⁷ de *media* en tiempo real.

Por otro lado, la PSTN fue diseñada específicamente para transmisiones de voz y desde el punto de vista técnico, satisface perfectamente la tarea; sin embargo desde un punto de vista de flexibilidad, ha quedado estancada debido a la rigidez de su tecnología. VoIP incorpora comunicaciones de voz en todos los protocolos de red existentes considerando su diseño, construcción y mantenimiento con base en las demandas especiales de las conversaciones de voz.

El problema con la transmisión de voz basada en paquetes mediante IP es que no es compatible con la manera en que habla el ser humano, el cual habla y escucha mediante la transmisión de *streams* de audio, mientras que los protocolos de Internet son diseñados para capturar la voz, encapsular los bits de información en paquetes y entregar cada paquete por cualquier camino posible hacia el punto final.

El mecanismo para transportar datos en una conexión VoIP generalmente involucra una serie de señales de transacción entre los clientes finales y las pasarelas intermedias, logrando dos *media streams* persistentes (una por cada dirección) para transportar la conversación.

Existen varios protocolos que proporcionan control y administración de sesión de telefonía en Internet, los cuales se conocen como protocolos de señalización o de establecimiento de llamada, y su función principal es establecer y terminar las llamadas en una red IP. Sus diferencias radican en que presentan redundancia en sus operaciones debido a su propio proceso de estandarización y a la variedad de soluciones propietarias que proporcionan los operadores de red, vendedores y usuarios.

La Figura 2.1 muestra el camino que siguen las señales y datos del procedimiento de establecimiento de llamada VoIP; los protocolos más utilizados son: H.323, MGCP, SIP e IAX, los cuales se describen a continuación.

¹⁷ Es un término que describe una estrategia sobre demanda para la distribución de contenido multimedia a través de Internet.

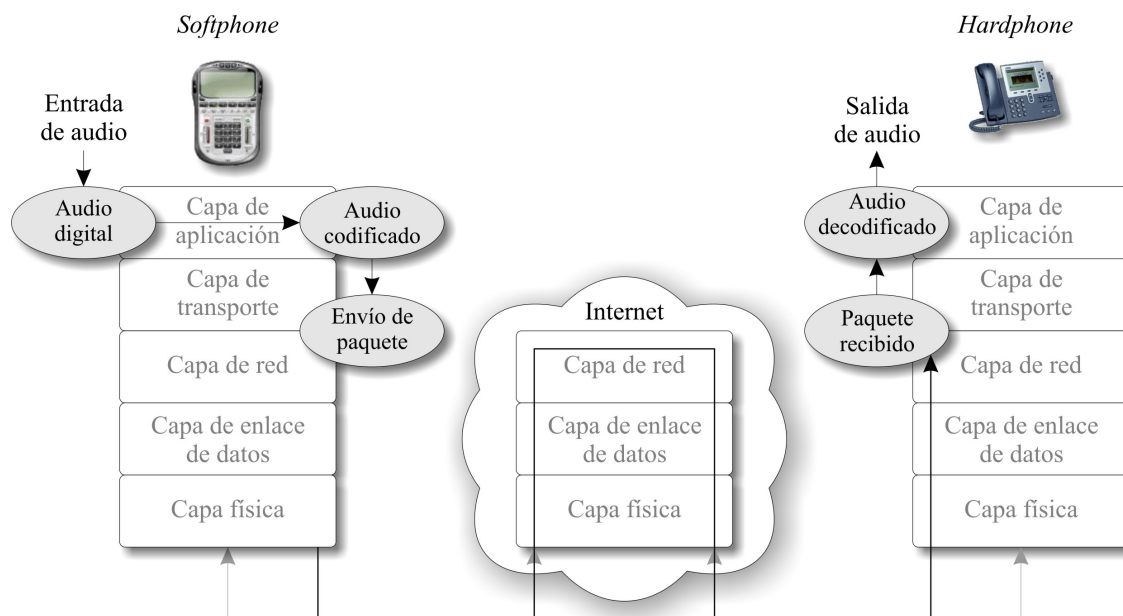


Figura 2.1. Procedimiento de establecimiento de llamada VoIP.

2.1. H.323

H.323 fue desarrollado por la ITU en mayo de 1996 para transmitir voz, video, datos y comunicaciones de fax a través de una red IP con conectividad a la PSTN. Desde entonces, el protocolo ha sufrido diversas modificaciones para adaptar su funcionamiento tanto en redes VoIP puras como en redes distribuidas. H.323 es el estándar *de facto* para aplicaciones de videoconferencias basados en IP [7].

H.323 es un sistema de procesamiento de llamada para voz, video y datos, que se ejecuta desde aplicaciones software que soportan conversaciones de voz (*softphones*), en PBXs, pasarelas, puentes (*bridge*) de conferencias, etc. Mediante H.323 se pueden encontrar recursos, registrar usuarios, asignar ancho de banda, negociar capacidad, configurar canales lógicos para el flujo de *media* (voz, fax, etc.), definir mensajes y formatos, configurar y terminar llamadas [3].

Las terminales de usuario H.323 soportan aplicaciones de tiempo real mediante dos caminos de audio, video o comunicaciones de datos, y pueden comunicarse con una pasarela H.323 o una unidad de control multipunto (MCU, *Multipoint Control Unit*).

Con H.323 se pueden realizar aplicaciones como: llamadas de voz, multi-conferencias, seminarios, sesiones de trabajo basadas en conversaciones con pizarra (*chalk talk*), fax y datos para sesiones unidifusión y multidifusión.

H.323 define las funciones de unidades de multiconferencia y utiliza diversos estándares de la ITU-T para administrar y controlar las sesiones de usuario, las cuales cubren los distintos aspectos que engloban una conversación de voz.

La Tabla 2.1 describe los protocolos que realizan las funciones de H.323, el cual no fue diseñado para trabajar con la arquitectura Web (HTTP, URLs y sintaxis de transferencia centrados en el texto), sus estructuras de datos y sintaxis de transferencia están basados en la capa de presentación del modelo OSI (capa 6).

La principal desventaja que presenta el protocolo H.323 es su complejidad, y aunque es uno de los protocolos con gran tráfico VoIP, los usuarios presentan nuevas necesidades respecto a los operadores de telefonía de las PSTN. Actualmente, existen empresas que hacen uso del protocolo H.323, por ejemplo Microsoft e IBM, sin embargo la mayoría está migrando hacia SIP o IAX.

Tabla 2.1. Funciones y protocolos que utiliza H.323.

Funciones	Protocolos
Direccionamiento	RAS (<i>Registration, Admission and Status</i>): el cual permite a una estación H.323 localizar estaciones H.323 mediante el <i>gatekeeper</i>
	DNS (<i>Domain Name Service</i>): servicio de resolución de nombres en direcciones IP con el mismo fin que el protocolo RAS pero a través de un servidor DNS
Señalización	Q.931: para señalización inicial de llamada
	H.225: para control de llamadas (señalización, registro y admisión, y paquetización/sincronización del <i>stream</i> de voz)
	H.245: para especificar mensajes de apertura y cierre de canales para <i>streams</i> de voz
Compresión de voz	Requeridos: G.711 y G.723
	Opcionales: G.728, G.729 y G.722
Transmisión de voz	UDP: la transmisión se realiza sobre paquetes UDP, pues aunque UDP no ofrece integridad en los datos, el aprovechamiento del ancho de banda es mayor que con TCP
	RTP (<i>Real Time Protocol</i>): maneja los aspectos relativos a la temporización, marcando los paquetes UDP con la información necesaria para la correcta entrega de los mismos en recepción
Control de la transmisión	RTCP (<i>Real Time Control Protocol</i>): para detectar situaciones de congestión de red y tomar acciones que corrijan la situación

2.1.1. Consideraciones de seguridad

H.323 es un protocolo relativamente seguro y no requiere muchas consideraciones de seguridad más allá de las comúnmente utilizadas por cualquier red que se comunica con Internet.

H.323 utiliza el protocolo RTP para comunicaciones de *media*, por lo que no soporta la trayectoria encriptada de los medios (*encrypted media paths*). La solución es el uso de una VPN o un túnel encriptado entre los puntos finales, siendo este último no siempre conveniente e incluso imposible de implementar.

2.1.2. H.323 y NAT

H.323 hace uso del protocolo RTP de la IETF para transportar *media* entre los puntos finales, con lo cual es similar a SIP respecto a las topologías de red que involucran a NAT (Apartado 2.3.2).

El método más simple es enviar al cliente interno los puertos apropiados del dispositivo NAT. Para recibir llamadas es necesario enviar el puerto TCP 1720 al cliente, los puertos UDP para RTP *media* y los canales de control RTCP. En casos particulares como un cliente MS Neetmeeting, es necesario enviar los puertos TCP mediante túneles H.245.

Es necesario el uso de un *gatekeeper*, ejecutándose en modo proxy, para dar soporte a clientes detrás de un dispositivo NAT. El *gatekeeper* requiere una interfaz unida a la subred privada para llevar el control del registro de los usuarios internos, asimismo se encarga de registrar a los clientes externos en el servidor *proxy*.

2.1.3. Arquitectura de H.323

La pasarela H.323 funciona como un nodo en una red de paquetes (una LAN por ejemplo) que se comunica con terminales H.323 u otras terminales ITU-T de otras redes de paquetes; si una de las terminales no es una terminal H.323, la pasarela se encarga de traducir el formato de transmisión entre las terminales (Figura 2.2) [3].

Una pasarela H.323 puede colaborar con otra pasarela H.323 y operar con otras redes ITU-T de conmutación de circuitos (SCN, *Switched-Circuits Network*), como por ejemplo [20]:

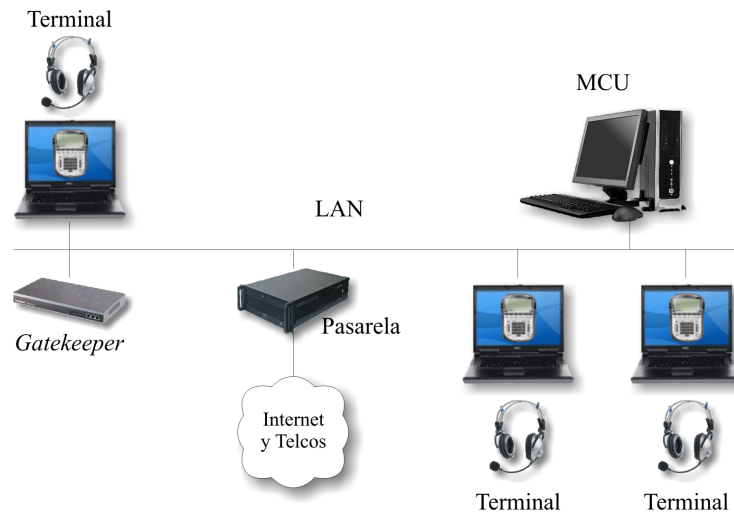


Figura 2.2. Arquitectura de H.323.

- Red de telefonía conmutada (GSTN, *General Switched Telephone Network*) o PSTN.
- Red digital de servicios integrados de banda estrecha (N-ISDN, *Narrowband-Integrated Services Digital Network*).
- Red digital de servicios integrados de banda ancha (B-ISDN, *Broadband-Integrated Services Digital Network*).

La pasarela puede operar como una unidad H.323 MCU, la cual soporta multiconferencia entre tres o más terminales y pasarelas. Dos terminales de conferencia punto a punto se pueden expandir a conferencias multipunto.

2.1.4. Operaciones H.323

H.323 invoca varias operaciones¹⁸ y protocolos para soportar comunicación entre el usuario final y los puntos finales (terminales, pasarelas o MCUs). La Figura 2.3 muestra las operaciones más importantes que pueden ser invocadas dependiendo de la implementación [3]:



Figura 2.3. Operaciones de H.323.

¹⁸ En la literatura de H.323 también se conocen como fases.

- *Evento 1 (Descubrimiento)*: Esta operación exige que un punto final busque un *gatekeeper* para registrarse, lo cual permite al administrador de la red controlar el uso de la red H.323, de forma que pueda permitir o restringir dicho registro. Durante esta operación, el punto final y el *gatekeeper* intercambian direcciones, reservando la dirección IP de multidifusión 224.0.1.41 para el descubrimiento del *gatekeeper*.
- *Evento 2 (Registro)*: Se define la forma en que un punto final se registra en un *gatekeeper*. Se utilizan las direcciones establecidas durante la operación de descubrimiento y se identifica el tipo del punto final (terminal de usuario final, pasarela o MCU). La idea básica de la operación de registro es permitir a los nodos H.323 ingresar a una zona de llamada, la cual es una parte de la red controlada por el *gatekeeper*. Se reservan los puertos UDP 1718 y 1719 para las operaciones de descubrimiento y registro, respectivamente.
- *Evento 3 (Configuración de conexión)*: Esta operación se encarga de configurar la conexión entre dos puntos finales para establecer la comunicación de voz.
- *Evento 4 (Capacidad de intercambio)*: El propósito de esta operación es asegurar que el punto final receptor reciba correctamente el tráfico multimedia enviado por el punto final emisor. Los puntos finales y el *gatekeeper* intercambian información sobre: sesión, velocidad de transferencia de datos y tipo de codec.
- *Evento 5 (Intercambio de canal lógico)*: Esta operación permite la transmisión de diferentes tipos de media *streams* sobre canales lógicos.
- *Evento 6 (Transferencia de carga útil)*: Una vez concluidas las operaciones anteriores, se puede intercambiar el tráfico de usuario.
- *Evento 7 (Finalización)*: Después de completar la sesión de usuario, se ejecutan las operaciones de finalización que solicitan la liberación de los canales lógicos y los recursos utilizados durante las operaciones anteriores.

La Figura 2.4 muestra el intercambio de mensajes para realizar la operación de descubrimiento del *gatekeeper*:

- *Evento 1*: El punto final envía al *gatekeeper* un mensaje de solicitud (GRQ) e inicializa un contador (*timer*).
- *Evento 2*: El *gatekeeper* examina la solicitud y puede o no responder, dependiendo de la implementación, con un mensaje de confirmación (GCF); si el punto final no recibe respuesta alguna, el contador expira, vuelve a enviar otro GRQ e inicializa nuevamente el contador; en caso de que continúe el problema, el administrador de red debe localizar la falla y solucionar el problema, alternativamente el *gatekeeper* debe regresar un mensaje de rechazo (GRJ) al punto final.



Figura 2.4. Operaciones de descubrimiento del *gatekeeper*.

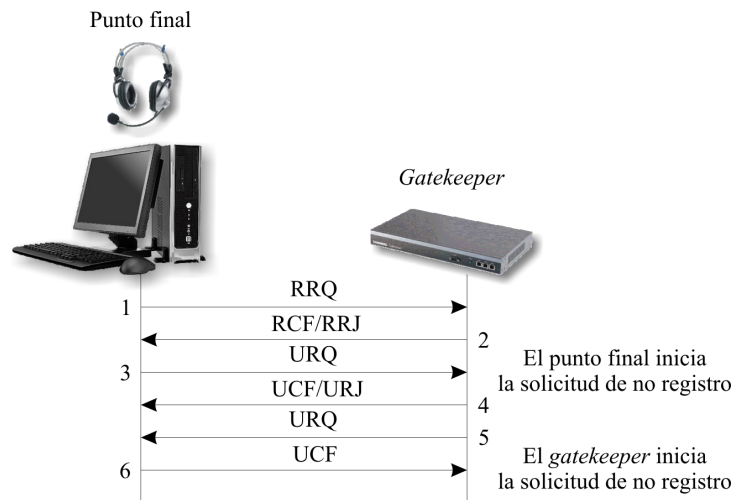


Figura 2.5. Procedimientos de registro.

Una vez que el proceso de descubrimiento se ha realizado con éxito, se inicializan las operaciones de registro, las cuales describen la forma en que un punto final proporciona al *gatekeeper* sus números de puertos de acceso y su(s) dirección(es) para registrarse en una zona de llamada.

La Figura 2.5 muestra el intercambio de mensajes entre el punto final y el *gatekeeper* realizados durante la operación de registro:

- *Evento 1:* El punto final envía al *gatekeeper* un mensaje de solicitud de registro (RRQ, *Registration Request*).
- *Evento 2:* El *gatekeeper* responde con un mensaje de confirmación de registro (RCF, *Registration Confirmation*) o con un mensaje de rechazo de registro (RRJ, *Registration Reject*).
- *Evento 3:* Tanto el punto final como el *gatekeeper* pueden cancelar el registro y terminar la conexión entre las dos entidades, para ello el punto final envía una petición de cancelación de registro (URQ, *Unregister Request*).
- *Evento 4:* El *gatekeeper* responde con un mensaje de confirmación de no registro (UCF, *Unregister Confirm*) o con un mensaje de rechazo de no registro (URJ, *Unregister Reject*).
- *Evento 5:* El *gatekeeper* inicia el proceso de cancelación de registro mediante el envío de un mensaje URQ.
- *Evento 6:* El punto final debe responder con un mensaje UCF.

2.1.5. Descripción general de H.323

H.323 define el uso del protocolo de señalización modificado Q.931. La Figura 2.6 muestra los mensajes H.323 utilizados entre dos puntos terminales (punto final 1 y punto final 2) y el *gatekeeper*. A continuación se describe la secuencia de eventos en donde cabe señalar que los puntos terminales intercambian mensajes Q.931 durante los eventos 3, 4, 7 y 8:

- *Evento 1:* El punto final 1 emite un mensaje de solicitud de admisión (ARQ) al *gatekeeper*.
- *Evento 2:* El *gatekeeper* regresa al punto final 1 un mensaje de confirmación de admisión (ACF) o de rechazo de admisión (ARJ) según sea el caso; asimismo se determina la naturaleza de la llamada, ya sea mediante conexión multipunto o punto a punto.
- *Evento 3:* El punto final 1 solicita el establecimiento de llamada con el punto final 2 a la vez que se realiza el proceso de configuración de los parámetros de llamada; se utiliza un identificador del punto final como medida de seguridad para asegurar que sea una terminal válida dentro de la zona de llamada.

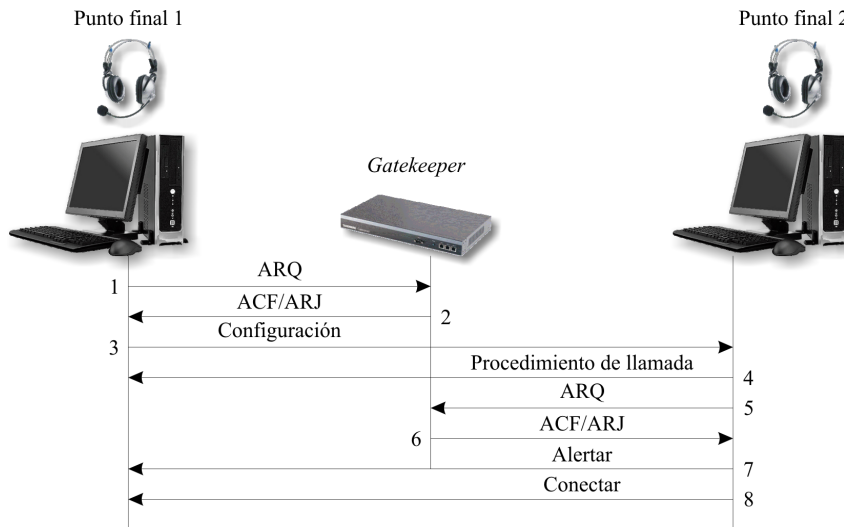


Figura 2.6. Admisión de procedimientos.

- *Evento 4:* El punto final 2 responde mediante el procedimiento de llamada; se otorgan direcciones para el punto final 1 y punto final 2 y éstas pueden ser direcciones E.164, identificadores H.323 o números de puertos.
- *Evento 5:* El punto final 2 envía un mensaje de solicitud de admisión (ARQ) al *gatekeeper*; y se solicita el ancho de banda necesario para realizar la llamada bidireccional.
- *Evento 6:* El *gatekeeper* regresa un mensaje ACF o ARJ, según sea el caso, al punto final 2.
- *Evento 7:* El punto final 2 alerta al punto final 1 sobre el establecimiento de llamada; se proporciona información sobre los protocolos opcionales de la serie Q (Q-Serie).
- *Evento 8:* El punto final 1 y el punto final 2 establecen la llamada mediante la concesión de un identificador único de conferencia.

2.2. MGCP

El protocolo del control de la entrada de los media (MGCP) fue desarrollado por la IETF y está definido en los RFCs 3435 y 2705, integra al protocolo simple del control de pasarelas (SGMP, *Simple Gateway Control Protocol*) y la especificación del protocolo de Internet para el control de dispositivos (IPDC, *Internet Protocol Device Control*) con la finalidad de simplificar el diseño de los dispositivos finales; establece la lógica para el establecimiento y control de llamadas que tienen que implementar las pasarelas de medios (*media gateways*) y los agentes de control de llamadas (*call agents*).

MGCP utiliza un modelo centralizado, en el cual los teléfonos MGCP no pueden llamar directamente a otros teléfonos MGCP sino que deben pasar a través de un controlador de llamadas [7].

MGCP describe una interfaz de programación y un protocolo complementario con el objetivo de definir las operaciones de las pasarelas de telefonía. El control de llamadas es realizado por los agentes de llamada, mientras que la pasarela de telefonía proporciona conversión y operaciones de internet (*internetworking*) entre las señales de audio utilizadas en telefonía de conmutación de circuitos, y los paquetes de datos utilizados por Internet u otras redes orientadas a conmutación de paquetes. El agente de llamada dirige las operaciones de la pasarela [3, 21].

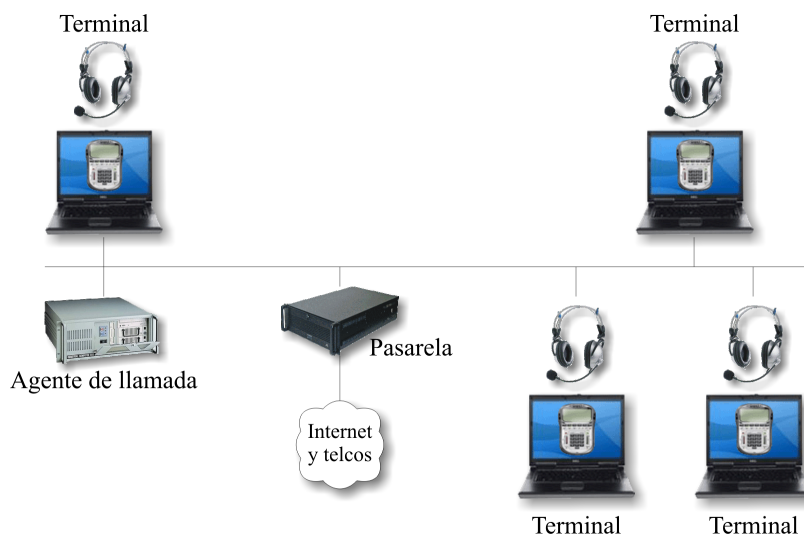


Figura 2.7. Arquitectura MGCP.

2.2.1. Arquitectura MGCP

La Figura 2.7 muestra la topología para el sistema de procesamiento de llamadas IP de MGCP. La arquitectura externa de MGCP es similar a la del protocolo H.248/Megaco (*Media Gateway Controller*), ambos sistemas utilizan el concepto de controlador de pasarela (*gateway controller*), soportan el procesamiento de llamadas y están basados en la Web; sin embargo, MGCP no utiliza los conceptos de terminación y de contexto que emplea Megaco.

2.2.2. Operaciones MGCP

La Figura 2.8 muestra los eventos correspondientes a un ejemplo de funcionamiento del MGCP con dos pasarelas, una residencial y una *trunking*; además se utiliza una base de datos común y una pasarela de control para identificar las operaciones realizadas, aunque realmente no se requieren en un procedimiento MGCP. A continuación se describen dichos eventos:

- *Evento 1:* El agente de llamada envía un mensaje de solicitud de notificación a la pasarela residencial antes de que ésta pueda manejar otra conexión, dicho mensaje no es una orden de configuración; los agentes de llamada y las pasarelas deben estar preconfigurados.
- *Evento 2:* La pasarela residencial acepta la solicitud mediante el envío de un mensaje ACK.
- *Evento 3:* El usuario y la pasarela residencial se ponen en estado de conectado (*off-hook*) para hacer una llamada.
- *Evento 4:* La pasarela residencial envía un mensaje de notificación codificado al agente de llamada para indicarle el estado de conectado.
- *Evento 5:* El agente de llamada reconoce el mensaje de notificación mediante el envío de un mensaje ACK.
- *Evento 6:* La respuesta del agente de llamada a la pasarela depende del tipo de línea supervisada; si es una línea convencional *dialup* (no dirigida), el agente de llamada envía a la pasarela un mensaje de solicitud de notificación para dar tono de marcado y recolectar dígitos.
- *Evento 7:* La pasarela responde al usuario con el tono de marcado y con un mensaje ACK al agente de llamada.
- *Evento 8:* La pasarela acumula los dígitos de acuerdo al mapa de dígitos enviado.
- *Evento 9:* Con base en el mapa de dígitos, la pasarela residencial notifica al agente de llamada los parámetros del evento observado, los cuales contienen la recolección de dígitos.

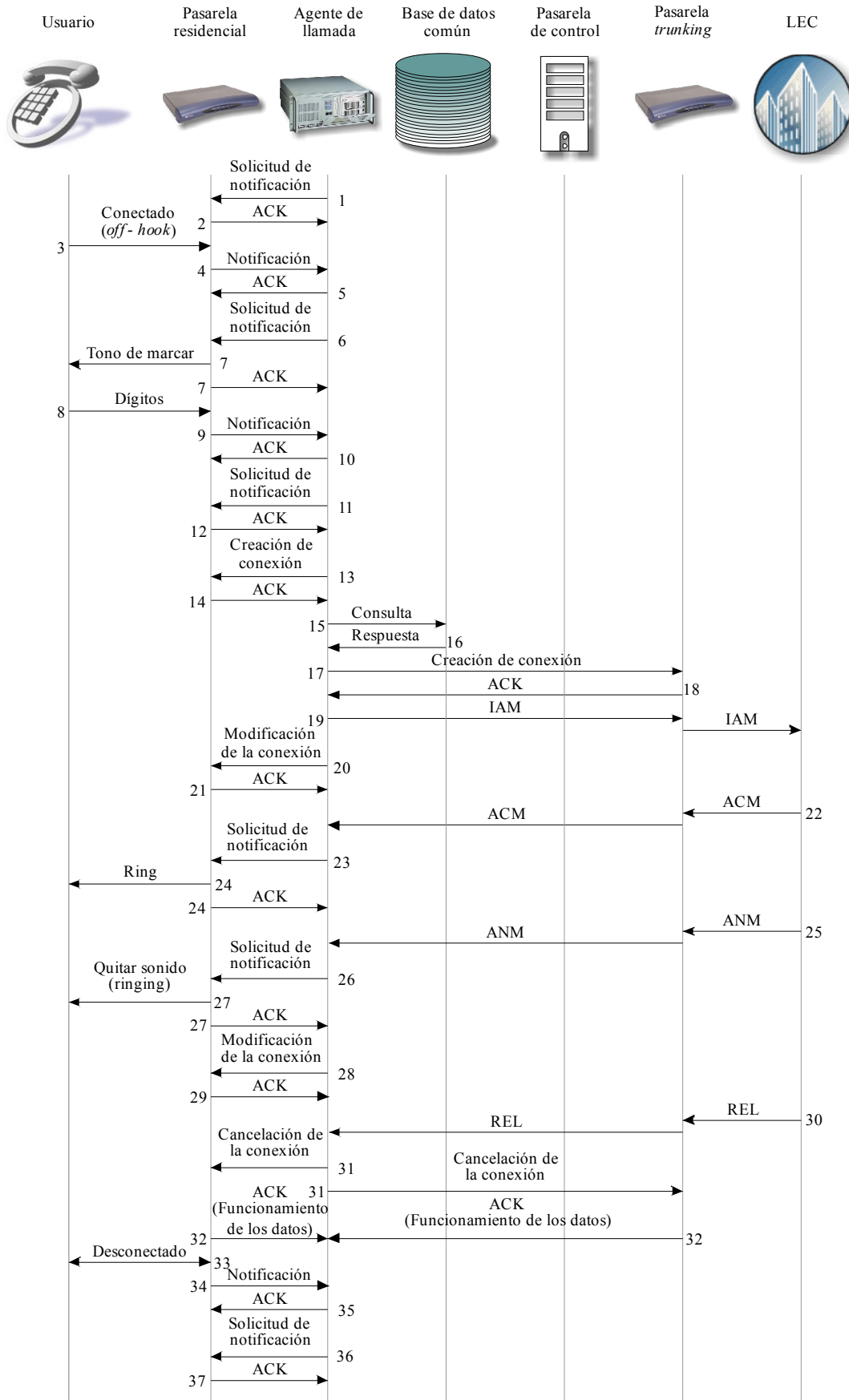


Figura 2.8. Flujos de eventos del protocolo MGCP.

- *Evento 10*: El agente de llamada envía un mensaje ACK a la pasarela residencial como reconocimiento del mensaje.
- *Evento 11*: El agente de llamada envía un mensaje de solicitud de notificación a la pasarela residencial para detener la recolección de dígitos y supervisar una transición de desconexión (*on-hook*).
- *Evento 12*: La pasarela residencial reconoce el mensaje de solicitud.
- *Evento 13*: El agente de llamada envía un mensaje de establecimiento de conexión para apoderarse del circuito entrante. Dicho mensaje contiene: periodo de paquetización en milisegundos, algoritmo de compresión (G.723, G.729, etc.), ancho de banda para la conexión y una bandera para utilizar o no la cancelación de eco.
- *Evento 14*: La pasarela residencial reconoce el mensaje mediante el envío de un mensaje ACK, el cual contiene la identificación de la conexión y la descripción de la sesión (SDP) que describe el tráfico de audio. Dicha descripción puede contener la dirección IP de la pasarela, el protocolo para transportar los paquetes (RTP), el puerto RTP (3456) y el perfil de audio-video (AVP, *Audio-Video Profile*¹⁹) de acuerdo al RFC 1890. Este mensaje también informa al agente de llamada que la pasarela está lista para soportar otros perfiles de audio.
- *Evento 15*: El agente de llamada consulta la base de datos común para determinar la dirección de la llamada y la pasarela de salida con la finalidad de establecer la conexión. La base de datos por lo regular contiene la información necesaria para establecer una ruta de llamada, que en su forma más simple es una tabla de ruteo IP.
- *Evento 16*: La base de datos regresa la información correspondiente al agente de llamada.
- *Evento 17*: El agente de llamada tiene la información para enviar un mensaje de establecimiento de conexión a la pasarela *trunking*. Dicho mensaje contiene los parámetros intercambiados en los eventos 13 y 14 entre la pasarela residencial y el agente de llamada.
- *Evento 18*: La pasarela *trunking* responde con un mensaje ACK, el cual contiene la SDP respecto a dirección IP, el número de puerto de acceso y el perfil RTP.
- *Evento 19*: El agente de llamada considera la información obtenida en los eventos anteriores para construir un mensaje SS7 ISUP de dirección inicial (IAM, *Inicial Address Message*) y lo envía a la pasarela *trunking*, la cual se encarga de transmitirlo a la portadora de intercambio local (LEC, *Local Exchange Carrier*).
- *Evento 20*: Se utiliza la información obtenida en el evento 18 para crear el mensaje de modificación de conexión que se envía a la pasarela residencial.
- *Evento 21*: La pasarela residencial acepta el mensaje.
- *Evento 22*: La LEC regresa al agente de llamada un mensaje SS7 ISUP de dirección completa (ACM, *Address Complete Message*), el cual contiene información que permite al agente de llamada dirigir las acciones de la pasarela residencial.
- *Evento 23*: Al recibir el ACM, el agente de llamada envía un mensaje de notificación de solicitud a la pasarela residencial, para que ésta ponga el tono de llamada en la línea.
- *Evento 24*: La pasarela residencial reconoce el mensaje y coloca el tono de llamada.
- *Evento 25*: Cuando la LEC responde a la llamada en el extremo remoto, envía un mensaje ISUP de respuesta (ANM, *Answer Message*) al agente de llamada.
- *Evento 26*: El agente de llamada envía un mensaje de solicitud de notificación a la pasarela

¹⁹ El AVP define el tipo de carga útil como puede ser G.711.

residencial para quitar el tono de llamada de la línea.

- *Evento 27*: La pasarela residencial acepta el mensaje y deshabilita el tono de llamada.
- *Evento 28*: La conexión en el extremo local se encuentra en modo de recepción, para cambiarla a modo *full-duplex*, el agente de llamada envía un mensaje de modificación de la conexión a la pasarela residencial.
- *Evento 29*: La pasarela residencial acepta y establece la conexión.
- *Evento 30*: Una vez que las partes hayan finalizado la conversación telefónica, la parte llamada cuelga y la condición conectado ocasiona el envío de un mensaje ISUP de liberación (REL, *Release Message*) al agente de llamada.
- *Evento 31*: El agente de llamada envía un mensaje a ambas pasarelas para eliminar la conexión, dicho mensaje contiene los identificadores correspondientes para que las pasarelas identifiquen la conexión a eliminar.
- *Evento 32*: Las pasarelas envían un mensaje de respuesta cuyos campos contienen información estadística de la llamada (duración, errores encontrados y número de paquetes VoIP intercambiados). En este evento, la pasarela de control se puede utilizar para almacenar el costo de la llamada.
- *Evento 33*: La parte local, que cuelga, pone en condición desconectado a la línea local.
- *Evento 34*: La pasarela residencial envía un mensaje de notificación al agente de llamada para indicarle la condición desconectado.
- *Evento 35*: El agente de llamada acepta el mensaje mediante el envío de un mensaje ACK.
- *Evento 36*: El agente de llamada reinicializa al punto final para informar a la pasarela la condición de conectado.
- *Evento 37*: La pasarela residencial acepta el mensaje y pone al punto final en el estado inicial en el que estaba antes de ocurrir el evento 1.

2.3. SIP

El protocolo de inicio de sesión (SIP) fue propuesto en febrero de 1996 por la IETF en el documento intitulado “draft-ietf-mmusic-sip-00”, dicho documento contenía especificaciones que difieren con las actuales y sólo contenía un tipo de petición para configuración de llamadas (*call setup request*).

En marzo de 1999, después de 11 revisiones, se presentó el RFC 2543 de SIP. Inicialmente fue ignorado debido a que H.323 era el protocolo de señalización para transporte de VoIP, sin embargo, SIP fue ganando popularidad gracias a que su especificación es de libre distribución y a su facilidad de uso e implementación comparado con H.323 [7].

SIP considera a cada extremo de la conexión como par (*peer*)²⁰ y se encarga de negociar la intercomunicación entre pares. SIP es un protocolo simple, con una sintaxis parecida a la de protocolos como HTTP (*Hypertext Transfer Protocol*) y SMTP (*Simple Mail Transfer Protocol*).

El mercado de tecnología VoIP ofrece una gran variedad de productos SIP debido a que su especificación soporta nuevas aplicaciones como transmisión de video, audio y cualquier tipo multimedia en tiempo real [7].

SIP es un protocolo de control y señalización cuyas principales funciones son establecer, modificar y finalizar sesiones entre usuarios. SIP soporta sesiones unidifusión, multidifusión, acopla-

²⁰ Un *host* o dispositivo que ocasiona la ejecución del protocolo.

miento de sesiones unidifusión, o una combinación de éstas; no actúa como pasarela de *medios* debido a que no soporta el transporte de cualquier *media stream*.

Su característica principal es la capacidad de soportar a usuarios móviles. Si un usuario registra su localización en un servidor SIP, éste dirige los mensajes al usuario o invocará operaciones *proxy* a otro servidor cercano a la localización del usuario. La capacidad móvil se aplica únicamente al usuario individual y no a la terminal del usuario (teléfono, computadora, etc.). Dichos aspectos diferencian a SIP de los otros protocolos de establecimiento de llamada.

SIP es una herramienta de apoyo para la telefonía IP debido a que:

- Puede operar sin estado (*stateless*)²¹ o con estado (*stateful*); una implementación sin estado proporciona buena escalabilidad debido a que una vez que se haya procesado la transacción, los servidores no mantienen la información sobre el estado de la llamada.
- Utiliza los formatos y la sintaxis del HTTP, lo cual proporciona una forma adecuada para operar con los buscadores actuales.
- El mensaje SIP es flexible en cuanto a su sintaxis y se puede describir de diversas formas, por ejemplo con MIME (*Multipurpose Internet Mail Extension*) o XML (*Extensible Markup Language*).
- Identifica al usuario mediante un localizador de recurso uniforme (URL, *Uniform Resource Locator*), con lo cual proporciona la capacidad de iniciar una llamada haciendo clic en una liga Web [3].

SIP soporta cuatro funciones: localización del usuario, determinación del tipo de *media* para la sesión, capacidad del usuario para participar en una sesión y establecimiento, transferencia y terminación de la llamada.

2.3.1. Consideraciones de seguridad

Respecto al proceso de autenticación de usuarios, SIP utiliza un sistema de reconocimiento/respuesta (*challenge/response*) basado en la siguiente secuencia: inicialmente el cliente origen envía una solicitud *INVITE* al servidor *proxy* destino; éste envía un mensaje de autorización 407 como respuesta, dicho mensaje contiene un conjunto de caracteres aleatorios (*nonce*), el cual se utiliza junto con la contraseña para generar la función hash MD5; en el siguiente envío de la primitiva *INVITE* se regresa dicha función. El cliente se autentifica sólo si ambas funciones, la que generó y la que recibe, son iguales.

Los ataques de negación del servicio (DoS, *Denial of Service*) son los más comunes en VoIP, y ocurren cuando se envía un gran número de solicitudes *INVITE* no válidas al servidor *proxy* con la finalidad de afectar el funcionamiento del sistema. Los DoS se pueden implementar fácilmente y sus efectos sobre los usuarios del sistema son inmediatos.

SIP proporciona diversos métodos para minimizar los efectos ocasionados por ataques DoS, sin embargo cada vez son más difíciles de prevenir. Por lo anterior, SIP implementa un mecanismo de seguridad en la capa de transporte (TLS, *Transport Layer Security*) entre el llamador y el dominio del llamado (*callee*), considerando las políticas de seguridad de la red local [7].

2.3.2. SIP y NAT

El mayor problema que ha tenido SIP es establecer transacciones a través de NAT, debido a que encapsula la dirección de la información en tramas de datos y NAT se ejecuta en una capa de red inferior; NAT no modifica la información de la dirección y desconoce los canales de *media* para

²¹ Hace referencia a un sistema o protocolo que no mantiene un estado persistente entre transacciones.

completar la conexión. Además los cortafuegos, normalmente integrados con NAT, no consideran los canales de *media* como parte de las transacciones SIP y bloquean la conexión [7].

2.3.3. Arquitectura SIP

Uno de los aspectos que diferencia a SIP de los demás protocolos de establecimiento de llamada es que no utiliza un controlador de pasarela, sino que se basa en un modelo Cliente/Servidor que contiene los siguientes elementos [3]:

- *Servidor*: Es un programa de aplicación que acepta mensajes de solicitud de otro programa llamado Cliente; una vez que el Servidor recibe los mensajes del Cliente, responde con un mensaje de respuesta.
- *Servidor proxy*: Actúa como Servidor y Cliente a la vez con el fin de realizar peticiones en nombre de otros Clientes. Puede mantener las peticiones internamente o también puede pasarlas a otros servidores. Un servidor *proxy* interpreta y puede reescribir un mensaje de petición SIP antes de enviarlo a otro Servidor o a un agente de usuario (UA, *User Agent*).
- *Servidor de redirección (redirect)*: Servidor que acepta peticiones SIP, identifica la dirección en la petición a una nueva dirección y regresa el mensaje al Cliente. Es diferente a un servidor *proxy* ya que no inicia su propia petición SIP y no envía peticiones SIP a otros Servidores y es diferente al UA porque no acepta llamadas.
- *Registrador*: Entidad SIP que acepta peticiones de registro y generalmente se hospeda en un servidor *proxy* o un servidor de redirección para ofrecer servicios de localización. El registrador se utiliza para registrar partes SIP en un dominio SIP, el cual es similar a la zona H.323 ya que es una entidad administrativa para un proveedor SIP.
- *Servidor de agente de usuario (UAS, User Agent Server)*: hace referencia a un servidor que entra en contacto con un usuario cuando se recibe una petición SIP y regresa una respuesta en nombre del usuario.
- *Agente de usuario (UA)*: Es una aplicación que contiene al cliente de agente de usuario (UAC, *User Agent Client*) y al servidor de agente de usuario (UAS, *User Agent Server*).

Como se muestra en la Figura 2.9, SIP consiste de dos componentes principales:

- *UA*: Es un sistema final que interactúa con el usuario y actúa como su representante; consiste de dos entidades:
 - *Cliente del protocolo (UAC)*: Inicia la llamada.
 - *Servidor del protocolo (UAS)*: Responde la llamada.

Debido a que el UA contiene tanto al UAC como al UAS, SIP puede operar como red par a par (*peer-to-peer*) mientras utilice el modelo Cliente/Servidor.

- *Servidor SIP*: Puede implementarse de dos formas:
 - *Servidor proxy*: Recibe una petición de un Cliente y decide a qué Servidor puede enviar la petición en caso de ser necesario; puede enviar la petición a otro Servidor: (servidor de redirección o UAS), la respuesta a la petición viajará a través de la misma trayectoria que la petición, pero en orden inverso. El servidor *proxy* actúa como Servidor y como Cliente para hacer peticiones en nombre de otros Clientes, interpreta los mensajes SIP y puede reescribir el mensaje de petición antes de enviarlo a otro Servidor o Cliente.
 - *Servidor de redirección*: No envía la petición sino que ordena al Cliente entrar en contacto directo con el siguiente Servidor; la dirección del mensaje SIP es mapeada en las nuevas direcciones y enviada de regreso al Cliente. Un servidor de redirección no puede actuar como Cliente.

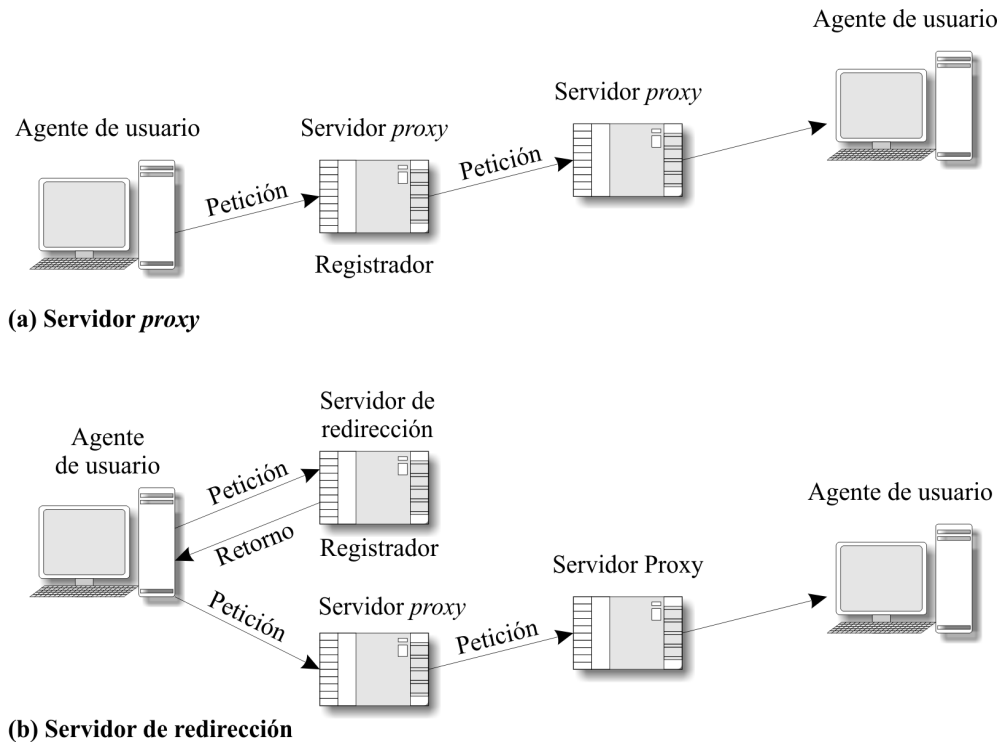


Figura 2.9. Clientes y servidores SIP.

2.3.4. Operaciones SIP

La Figura 2.10 muestra un ejemplo de las operaciones de SIP:

- *Evento 1:* Joan Gillen (*kgillen@earthlink.net*) está intentando comunicarse con la UTM (dirección *utm@utm.mx*). El servidor *proxy* recibe un mensaje INVITE (este mensaje viene del agente de usuario, el cual está alojado en el host de Joan Gillen).
- *Evento 2:* El servidor *proxy* pregunta a un servidor de localización para descubrir más información sobre la parte llamada (*utm@utm.mx*).

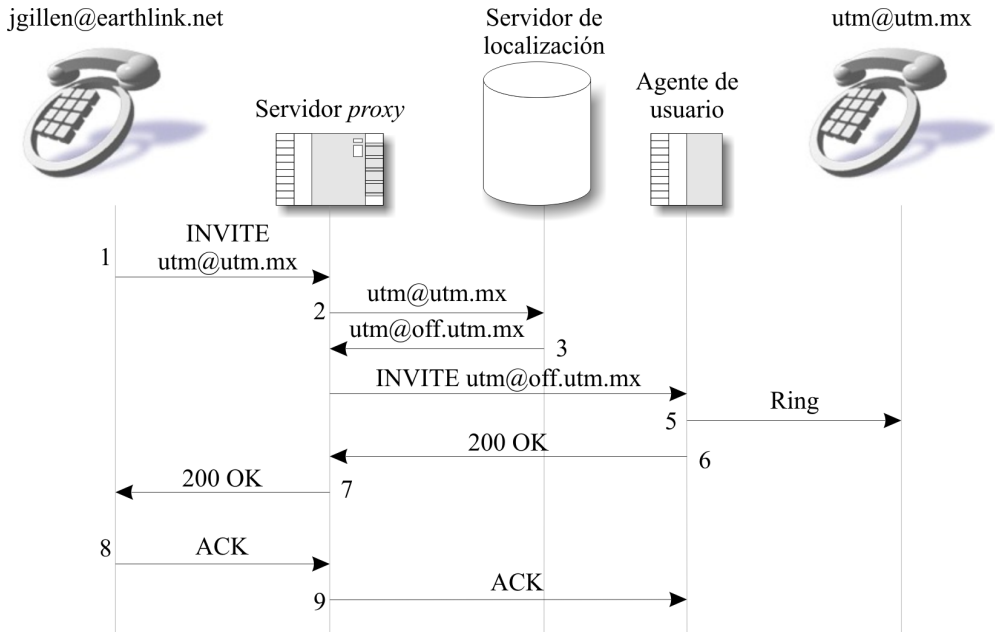


Figura 2.10. Operaciones SIP.

- *Evento 3*: El servidor de localización responde y proporciona un nombre más exacto (y una localización cuando el nombre se asocia a una dirección IP). Las operaciones SIP hacen uso del DNS.
- *Evento 4*: Dicha información se utiliza por el servidor *proxy* para enviar el INVITE (con la nueva información) al UA que mantiene el nombre (utm@utm.mx).
- *Evento 5*: El agente de usuario alerta a la parte llamada con un tono de llamada.
- *Evento 6*: El agente de usuario envía de regreso un mensaje SIP ACK OK, con el código 200 para aceptar e indicar que se está procesando la petición de llamada.
- *Evento 7*: El servidor *proxy* envía un mensaje ACK OK a la parte que llama.
- *Evento 8*: Para completar la operación, la parte que llama envía un mensaje SIP ACK al servidor *proxy*.
- *Evento 9*: El servidor *proxy* retransmite el mensaje SIP ACK al UA.

2.4. Protocolo IAX

IAX (*Inter-Asterisk Exchange*) es un protocolo desarrollado por la firma Digium con la finalidad de comunicar servidores VoIP; soporta una gran variedad de codecs y un gran número de canales (*streams*), con lo que se puede utilizar para transportar señalización y cualquier tipo de datos entre puntos finales (terminales VoIP) a través del puerto UDP 4569. IAX es un protocolo binario, diseñado y organizado para reducir la carga en flujos de datos de voz.

IAX soporta el envío de señalización y datos por múltiples canales, con lo cual los datos de varias llamadas se encapsulan en un conjunto de paquetes y se añaden a un datagrama IP, reduciendo el retardo y el *overhead*²² asociado a los canales individuales, lo anterior se conoce como *trunking*²³ y ayuda a mejorar la utilización del ancho de banda y reducir los tiempos de procesamiento [7].

IAX proporciona control y transmisión de flujos de datos multimedia sobre redes IP, cuyas principales aplicaciones son videoconferencias y presentaciones remotas. Por otro lado, IAX es un protocolo transparente a los cortafuegos y eficaz para trabajar en redes internas debido a que el tráfico de voz se transmite en banda (*in-band*).

El protocolo IAX establece sesiones internas que pueden utilizar cualquier codec para transmisión de voz o video, y está basado en los estándares SIP, MGCP y RTP (*Real-Time Transfer Protocol*).

El protocolo IAX fue diseñado para transmitir voz pero puede transportar cualquier *media stream*, incluyendo video. Actualmente IAX es un protocolo abierto y la comunidad de desarrolladores de tecnología IAX están incorporando diversos tipos de *media*.

2.4.1. IAX y NAT

El protocolo IAX2 fue diseñado para trabajar en dispositivos NAT. El uso de un puerto normal UDP para señalización y transmisión de comunicación mantiene los requisitos mínimos que exigen los cortafuegos, lo cual facilita la implementación de IAX en redes seguras [7].

²² Incremento de ancho de banda debido a la información que se añade a cada paquete para su identificación, control de flujo, etc.

²³ Enlace de comunicaciones entre dos puntos. IAX permite el intercambio de múltiples medias entre dos pares mediante multiplexación de la carga útil de los media (*payload media*) en un único enlace de llamada, con la finalidad de reducir la utilización del ancho de banda. Lo anterior puede ocurrir en una o ambas direcciones del intercambio IAX.

2.4.2. Descripción del protocolo IAX

IAX es un protocolo par a par orientado a VoIP que incluye funciones de control y de *media*, diseñado para describir y transportar llamadas multimedia mediante el IP.

El diseño de IAX permite la multiplexación de señales y llamadas multimedia sobre un mismo puerto UDP asociado entre dos pares. La señalización unificada de IAX y la trayectoria de *medias* logran transparencia sobre NAT, la cual es una ventaja de IAX sobre otros protocolos similares.

IAX es un protocolo binario cuyos principales beneficios son la eficiencia en la asignación del ancho de banda, robustez contra ataques y facilidad de implementación.

La unidad elemental de comunicación entre dos pares IAX es una trama (*frame*); IAX define varias clases de tramas [19]

- *Tramas completas (full frames)*: Conocidas como tramas F; transportan la señalización y el control de los datos e incluyen elementos de información (IEs, *Information Elements*)²⁴ que describen varios tipos de datos de usuarios o llamadas específicas.
- *Meta tramas (meta frames)*: Destinadas a establecer llamadas o la transmisión de video *streams*, se dividen en tramas meta video (*meta video frames*) y en meta tramas *trunk* (*meta trunk frames*).
- *Mini tramas (mini frames)*: Conocidas como tramas M, transportan datos *media stream* con una cabecera pequeña.

Una llamada basada en IAX puede consistir de varios segmentos o enlaces de llamadas (*legs*), los cuales se pueden implementar mediante diferentes protocolos. IAX es responsable del establecimiento de uno o varios enlaces de llamadas dentro de la trayectoria completa.

IAX es un protocolo par a par optimizado; si el par intermediario de dos enlaces de llamadas adyacentes decide no permanecer en la trayectoria de la llamada, puede solicitar un cambio en la trayectoria y darse de baja, de tal manera que la trayectoria se establece una vez que todos los pares confirmen que pueden comunicarse correctamente. IAX soporta métodos múltiples de autenticación de usuario y autorización como mecanismos de seguridad.

2.4.3. Indicador uniforme del recurso IAX

Un indicador uniforme del recurso (URI, *Uniform Resource Identifier*) identifica un recurso de comunicación válido para la versión 2 de IAX (IAX2). Un URI IAX contiene la suficiente información para iniciar una llamada; los URIs se asocian a los recursos del servidor al que se dirigen las llamadas.

Un URI IAX tiene la estructura `iax2:[username@]host[:port][/number[?context]]`, donde [19]:

- *Iax2*: Identifica el uso del protocolo IAX en su versión 2.
- *Username*: Cadena de caracteres utilizada para propósitos de identificación.
- *Host*: Dominio del recurso; éste puede contener un nombre de dominio (DNS) o una dirección IPv4 o IPv6.
- *Port*: Número de puerto UDP de acceso.
- *Number*: Nombre o número que identifica el recurso en el *host*.
- *Context*: Nombre de la partición del *host* en la cual se identifica o procesa el servicio.

²⁴ Unidad discreta de datos que se agrega a la trama IAX para especificar datos específicos de la llamada.

2.4.4. Transporte de mensajes IAX

Los mensajes IAX se dividen en dos categorías [19]:

- *Confiables*: Se refieren a las tramas completas, a las cuales se les pueden asociar atributos adicionales (por ejemplo IEs).
- *No garantizados*: Se refieren a las mini tramas y meta tramas, las cuales contienen el identificador de llamada origen y pueden no tener elementos de información.

Todos los mensajes son confiables, excepto los de voz y video, y son transportados por un esquema que mantiene contadores para cada mensaje y un sello de tiempo (*timestamp*) para ambos pares involucrados en la llamada. Las cuentas son por llamada; cada par mantiene un temporizador (*timer*) para todos los mensajes confiables y periódicamente debe retransmitirlos hasta que sean reconocidos o reintentarlo cuando se haya excedido el tiempo límite.

Cuando se inicia una llamada, los números de secuencia entrantes y salientes de los mensajes se deben fijar a cero; cada mensaje confiable enviado incrementa al contador de mensajes en uno, excepto los mensajes ACK, INVAL, TXCNT, TXACC y VNAK; el mensaje incluye el contador de mensajes de salida y el número más alto del mensaje de entrada que ha sido recibido, además, contiene un sello de tiempo que representa el número, en milisegundos, desde que dio inicio la llamada, y en el caso de ciertos mensajes de sincronización de red, se añade una copia del sello de tiempo recibido, la cual puede ser aproximada pero debe estar en orden.

Cuando se recibe un mensaje, se debe revisar el sello de tiempo para asegurar el orden de transmisión; si un mensaje se recibe fuera de orden, éste debe ser ignorado y se envía un mensaje VNAK para resincronizar los pares; si el mensaje es un mensaje confiable, se utiliza el contador de mensajes de entrada para reconocer todos los mensajes hasta el número de secuencia que ha sido enviado; si no se obtiene reconocimiento alguno, por defecto el enlace se debe considerar inutilizable y se debe dar por terminada la llamada por dicho enlace.

El protocolo IAX define dos temporizadores [19]:

- *Retransmisión*: En cada llamada existe un temporizador o contador de tiempo para saber cuánto tiempo habrá que esperar para el reconocimiento del mensaje. El periodo máximo de tiempo es de 10 segundos.
- *Periodo de registro*: Los registros son validos por un periodo de tiempo específico. Es responsabilidad del Cliente renovar el registro antes de que expire el periodo de tiempo.

Además, en la capa de aplicación define al temporizador de llamada y al temporizador de marcado (*timer ring*).

En los siguientes incisos se describen los principales mensajes IAX.

2.4.4.1. Registro

Para que un cliente IAX pueda comunicarse con otro cliente IAX, el *registrant*²⁵ debe conocer la dirección de red del *registrar*²⁶, la cual se obtiene de forma manual o mediante la consulta a un directorio. IAX permite que un usuario pueda autenticarse de diferentes maneras (Apartado 2.4.9).

La Tabla 2.2 muestra los mensajes que IAX define para realizar el registro del par.

²⁵ Es un par que realiza peticiones (REGISTER) para anunciar la dirección de un recurso.

²⁶ Es un par que procesa la petición (REGISTER) y coloca la información que recibe en el servicio de localización.

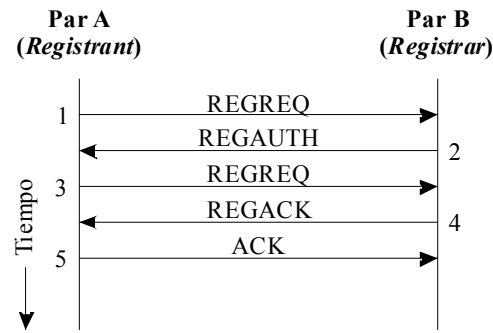


Figura 2.11. Secuencia de solicitud de registro IAX.

Tabla 2.2. Mensajes utilizados para el registro del par.

Mensaje	Definición	IEs
REGREQ	Solicitud de registro	<i>Username, Password, MD5 Result</i>
REGAUTH	Solicitud de autenticación	<i>Username, Auth methods, challenge</i>
REGACK	Reconocimiento de registro	<i>Username, Date Time, Apparent Address, Message Count, Calling Number, Calling Name, Firmware Version y Refresh</i>
ACK	Reconocimiento	-
REGREL	Liberación de registro	<i>Username, Password, MD5 Result, RSA Result, Cause y Cause Code</i>
REGREJ	Rechazo de registro	<i>Cause, Cause Code</i>

A continuación se describe la secuencia de mensajes utilizados para realizar el registro de dos pares (Figura 2.11):

- *Evento 1:* El registro inicia cuando el par A (*Registrant*) envía un mensaje REGREQ al par B (*Registrar*).
- *Evento 2:* Si se requiere realizar el proceso de autenticación, el par B responde con un mensaje REGAUTH indicando los métodos de autenticación que soporta.
- *Evento 3:* El par A reenvía un mensaje REGREQ con los datos solicitados en el proceso de autenticación.
- *Evento 4:* Si el método de autenticación es correcto, el par B envía un mensaje REGACK con la dirección de identificación y el tiempo de actualización/expiración.
- *Evento 5:* El par A acepta las condiciones de registro mediante el envío de un mensaje ACK.

En cualquier momento del proceso de registro, el par B puede enviar un mensaje REGREJ para rechazar el registro. IAX define un tiempo de registro de 60 segundos, el cual debe ser asumido por ambos pares. Si el proceso de registro no se lleva a cabo en dicho periodo, el par A puede actualizar el proceso de registro nuevamente mediante la secuencia descrita anteriormente. Si el par A decide suspender el proceso de registro, envía un mensaje REGREL para forzar la expiración del registro con el par B.

Un par A puede también forzar una expiración en el par B enviando un mensaje REGREL, este mensaje puede ser reemplazado con un REGAUTH o si las suficientes credenciales fueron incluidas, será aceptado con REGACK. En respuesta a un mensaje REGAUTH, se debe enviar un REGREL con las credenciales especificadas.

La Figura 2.12 muestra la secuencia de mensajes utilizados en la liberación del registro:

- *Evento 1:* El par A envía un mensaje REGREL para forzar el tiempo de expiración de registro en el par B.

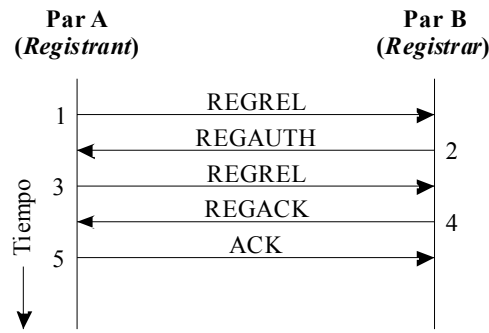


Figura 2.12. Secuencia de liberación del registro IAX.

- *Evento 2:* El par B envía un mensaje REGAUTH como respuesta al mensaje REGREL.
- *Evento 3:* El par A envía un mensaje REGREL con las credenciales solicitadas.
- *Evento 4:* Se envía un mensaje REGACK reconociendo de la liberación de registro.
- *Evento 5:* Se envía un ACK en reconocimiento del mensaje REGACK recibido.

2.4.4.2. Administrador de enlaces de llamada

El protocolo IAX permite configurar enlaces de llamada (*call legs*) entre dos pares con el objetivo de establecer una llamada. Para ello la Tabla 2.3 muestra los mensajes definidos por IAX.

Tabla 2.3. Mensajes utilizados en la administración de la llamada.

Mensaje	Definición	IEs
NEW	Iniciar una llamada	<i>Version, Called Number, Auto Answer, Codec Pref, Calling Presentation, Calling Number, Calling TON, Calling TNS, Calling Name, ANI, Language, DNID, Called Context, Username, Password, Format, Capability, ADSICPE, Date Time</i>
ACCEPT	Aceptación	<i>Format</i>
REJECT	Rechazo de solicitud	<i>Cause y Cause Code</i>
HANGUP	Terminación de una llamada	<i>Cause y Cause Code</i>
AUTHREP	Respuesta de autenticación	<i>RSA Result, MD5 Result y Password</i>
AUTHREQ	Solicitud de autenticación	<i>Username, Auth Methods, Challenge</i>

El proceso inicia cuando un par envía un mensaje NEW indicando el número o nombre del par destino (*called party*). El par destino puede responder con un mensaje de solicitud de autenticación (AUTHREQ), de rechazo de solicitud (REJECT) o de aceptación (ACCEPT).

El mensaje AUTHREQ indica los esquemas de autenticación permitidos y da lugar al envío de un mensaje AUTHREP. El mensaje REJECT indica que la llamada no puede ser establecida en ese momento. El mensaje ACCEPT indica que se ha establecido el enlace entre ambos pares y se envía después de haber recibido un mensaje NEW y cuando ha ocurrido la autenticación. Una vez establecido el enlace, se puede proceder con la señalización de la llamada.

2.4.4.3. Control de llamada

Los mensajes IAX para el control de llamada proporcionan funciones de señalización punto a punto comunes a otros protocolos de control de telefonía. La Tabla 2.4 muestra los principales mensajes para el control de llamada, los cuales deben ser enviados una vez que un enlace de llamada IAX ha sido aceptado.

Tabla 2.4. Mensajes utilizados en el control de llamada.

Mensaje	Definición
PROCEEDING	Indica que la solicitud está siendo atendida pero que aún no alcanza al par destino de la llamada
RINGING	Indica que se ha procesado el mensaje de petición de llamada y alerta sobre un posible establecimiento de llamada entre dos pares
ANSWER	Indica la aceptación de una petición de llamada

2.4.4.4. Operaciones de conexión entre llamada

El protocolo IAX cuenta con mensajes que pueden enviarse entre pares una vez que se ha establecido una llamada. La Tabla 2.5 muestra los mensajes IAX utilizados en la conexión entre llamada (*mid-call*).

Tabla 2.5. Mensajes utilizados en las operaciones de conexión entre llamada.

Mensaje	Definición	IEs
FLASH	Indica una característica entre llamada de dispositivos análogos	-
HOLD	Ordena al sistema remoto que detenga la transmisión de audio sobre el canal y opcionalmente reemplace el audio con música u otros sonidos	-
UNHOLD	Continuar con la transmisión de audio sobre el canal previamente detenido	-
QUELCH	Indica al sistema remoto que detenga la transmisión de audio en el canal	-
UNQUELCH	Indica al sistema remoto que reactive la transmisión de audio en el canal	-
TRANSFER	Transferir una llamada	<i>Called Number, Called Context</i>

2.4.4.5. Optimización de la trayectoria de llamada

Si un par está administrando una llamada entre otros dos pares y no está interesado en monitorear el progreso, contenido o duración de dicha llamada, puede indicar a los otros dos pares que se comuniquen directamente mientras él deja de administrar la llamada.

Lo anterior se conoce como optimización de la trayectoria de la llamada o supervisión de transferencia, con lo cual se asegura que la llamada no se pierda durante el progreso; el par que administra no cede el control del proceso de la llamada hasta asegurarse que los otros dos pares se están comunicando. La Tabla 2.6 lista los mensajes IAX para realizar dicho procedimiento.

Tabla 2.6. Mensajes utilizados en la optimización de la trayectoria de la llamada.

Mensaje	Definición	IEs
TXREQ	Utilizado para iniciar el proceso de transferencia	<i>Apparent Address, Call Number, Transfer ID</i>
TXCNT	Verifica conectividad con un par de reemplazo	<i>Transfer ID</i>
TXACC	Verifica conectividad con un par de reemplazo	<i>Transfer ID</i>
TXREADY	Indica el establecimiento de conectividad con el par que recibió instrucciones de transferir la llamada	<i>Transfer ID</i>
TXREL	Indica que el proceso de transferencia ha sido completado exitosamente	<i>Call Number</i>
TXREJ	Indica que la transferencia no puede proceder	-

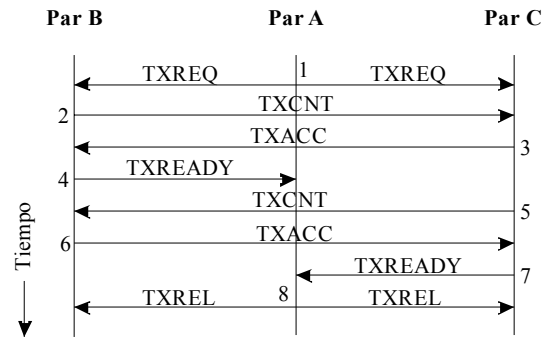


Figura 2.13. Secuencia para la optimización de la trayectoria de la llamada.

La Figura 2.13 muestra la secuencia de mensajes utilizados para optimizar la trayectoria de una llamada:

- *Evento 1:* El par A envía un mensaje TXREQ a los pares B y C para iniciar el proceso de transferencia de la trayectoria de la llamada.
- *Evento 2:* El par B envía un mensaje TXCNT al par C para verificar la conectividad.
- *Evento 3:* El par C responde con un mensaje TXACC para confirmar la conectividad.
- *Evento 4:* El par B envía un mensaje TXREADY al par A para indicar que ha verificado la conectividad con el par C.
- *Evento 5:* El par C envía un mensaje TXCNT al par B para verificar la conectividad.
- *Evento 6:* El par B envía un mensaje TXACC al par C para confirmar la conectividad.
- *Evento 7:* El par C envía un mensaje TXREADY al par A para indicar que existe conectividad con el par B.
- *Evento 8:* El par A envía un mensaje TXREL a los pares B y C para indicar que el proceso de transferencia de la trayectoria ha sido completado.

2.4.4.6. Finalización de llamada

El proceso para terminar una llamada varía dependiendo del proceso particular que la llamada esté ejecutando; se utilizan los mensajes HANGUP, REJECT, TRANSFER y TXREADY.

2.4.4.7. Supervisión o monitoreo de la red

El protocolo IAX cuenta con herramientas para determinar la carga de la red, las cuales se basan en los mensajes de la Tabla 2.7.

Tabla 2.7. Mensajes IAX para el monitoreo de la red.

Mensaje	Definición
POKE	Enviado para probar la conectividad de un par remoto cuando no está establecida una llamada
PING	Enviado para probar la conectividad de un par remoto durante el establecimiento de una llamada
PONG	Respuesta a un mensaje PING o POKE, y se encarga de reconocer la conexión
LAGRQ	Determina el retraso entre dos pares, incluyendo el que introduce la memoria temporal (<i>jitter buffer</i>)
LAGRP	Es una respuesta de retraso, enviado en respuesta al mensaje LAGRQ

La Figura 2.14(a) muestra la secuencia de mensajes para monitorear la red.

- *Evento 1:* El par A envía un mensaje PING para comprobar la conectividad con el par B.
- *Evento 2:* El par B responde con un mensaje PONG para reconocer la conectividad.

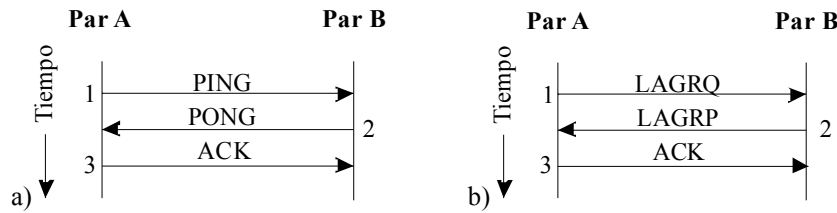


Figura 2.14. Secuencia de mensajes para monitorear la red y para determinar el retardo entre dos pares.

- *Evento 3:* Al recibir la respuesta, el par A responde con un mensaje ACK para indicar que está enterado del establecimiento de conexión con el par B.

La Figura 2.14(b) muestra la secuencia de mensajes para determinar el retardo entre dos pares:

- *Evento 1:* El par A envía un mensaje LAGRQ al par B para determinar el retardo existente entre ambos pares.
- *Evento 2:* El par B responde con un mensaje LAGRP para indicar el retardo existente.
- *Evento 3:* El par A envía un mensaje ACK para indicar que ha recibido la información.

2.4.4.8. Dígito de marcado

El soporte de dígito de marcado es una parte opcional del protocolo IAX y está diseñado para soportar dispositivos que no mantienen sus propios planes de marcación, por ejemplo los adaptadores de teléfonos analógicos (ATA, *Analog Telephone Adapter*). La Tabla 2.8 muestra los mensajes para marcación.

Tabla 2.8. Mensajes IAX para marcación.

Mensaje	Definición	IEs
DPREQ	Petición al servidor para analizar el número llamado y determinar si en el par remoto existe un patrón de marcado (<i>dialplan</i>) válido	<i>Call Number</i>
DPREP	Respuesta al mensaje DPREQ. Contiene el estado del patrón de marcado de entrada solicitado en el IE <i>Call Number</i> del mensaje DPREQ	<i>Call Number, Dial plan Status, Dial plan Refresh</i>
DIAL	Para marcar a un número	<i>Called Number, Called Context</i>

2.4.4.9. Descarga de *firmware*

El protocolo IAX permite a los dispositivos descargar sus controladores *firmware*; cabe mencionar que dicho proceso carece de seguridad. La Tabla 2.9 muestra los mensajes IAX que se utilizan en la descarga de controladores.

Tabla 2.9. Mensajes IAX utilizados en la descarga de controladores *firmware*.

Mensaje	Definición	IEs
FWDOWNL	Petición realizada por un dispositivo IAX para descargar su controlador <i>firmware</i> binario	<i>DEVTYPE, FWBLOCKDATA</i>
FWDATA	Respuesta al mensaje FWDOWNL. Consiste en un bloque <i>firmware</i> y los IEs para identificarlo	<i>FWBLOCKDESC, FWBLOCKDATA</i>

La Figura 2.15 muestra la secuencia de mensajes para descargar los controladores *firmware*:

- *Evento 1:* El par A envía una petición para descargar un *firmware* binario.

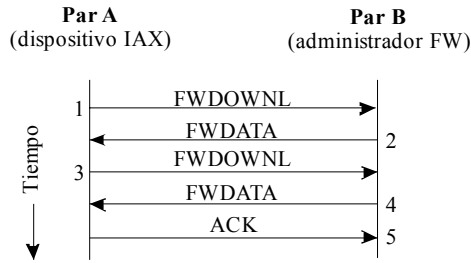


Figura 2.15. Secuencia de mensajes para descargar los controladores *firmware*.

- *Evento 2*: El par B envía el bloque del *firmware* y los IEs necesarios para identificar el *firmware* que está siendo enviado.
- *Evento 3*: Si el IE *FWBLOCKDATA* no es de longitud 0 se envía otro mensaje *FWDOWNL* por parte del par A solicitando el siguiente bloque del *firmware* a descargar.
- *Evento 4*: El par B responde al mensaje recibido enviando el bloque del *firmware* y los IEs necesarios para identificar el *firmware* que está siendo enviado.
- *Evento 5*: Si el IE *FWDATA* es el último de la transacción, significa que el IE *FWBLOCKDATA* es de longitud 0, y por lo tanto se envía un *ACK* como reconocimiento.

2.4.4.10. *Aprovisionamiento*

El protocolo IAX permite aprovisionamiento de dispositivos pares, consistente en especificar información de configuración para un dispositivo sin estado. El aprovisionamiento permite configurar la información de la red, del usuario y del codec a utilizar.

La Tabla 2.10 muestra el mensaje IAX para realizar el aprovisionamiento; el cual puede ser configurado para un tipo de dispositivo en particular (*device type*) o para un dispositivo específico (*service identifier*).

Tabla 2.10. Mensaje de aprovisionamiento.

Mensaje	Definición	IEs
PROVISION	Un mensaje PROVISION es enviado por un par de aprovisionamiento a un dispositivo IAX remoto	<i>PROVISIONING, AESPROVISIONING</i>

La Figura 2.16 muestra la secuencia de mensajes para realizar el aprovisionamiento de dispositivos IAX:

- *Evento 1*: El servidor de aprovisionamiento envía un mensaje *PROVISION* a un dispositivo IAX con la finalidad de especificar información de configuración del dispositivo que no mantiene su estado.
- *Evento 2*: El dispositivo IAX responde con un mensaje de aceptación de aprovisionamiento.
- *Evento 3*: El servidor de aprovisionamiento responde con un mensaje *ACK* para reconocer la aceptación de aprovisionamiento por parte del dispositivo IAX.

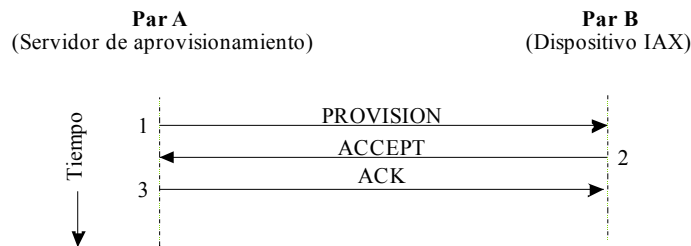


Figura 2.16. Secuencia de mensajes para el aprovisionamiento de un dispositivo IAX.

2.4.4.11. Misceláneos

El protocolo IAX cuenta con diversos mensajes, conocidos como misceláneos, para resolver contingencias ocasionadas durante el envío y recepción de mensajes IAX (Tabla 2.11).

Tabla 2.11. Mensajes misceláneos.

Mensaje	Definición	IEs
ACK	Mensaje de reconocimiento enviado sobre la recepción de una trama completa	-
INVAL	Enviado como respuesta a un mensaje recibido que no es valido	-
VNAK	Enviado cuando un mensaje es recibido fuera de orden, particularmente cuando una mini trama es recibida antes de la primer trama completa de voz, en una llamada. Es una petición de retransmisión de mensajes perdidos	-
MWI	Utilizado para indicar a un par remoto que esta esperando uno o mas mensajes	<i>MSGCOUNT</i>
UNSUPPORT	Enviado en respuesta a un mensaje que no es soportado por un par IAX	<i>UNKNOWN</i>

2.4.4.12. Mensajes media

El protocolo IAX define a las tramas M para transportar audio y video, sin embargo, cada vez que el sello de tiempo (*time-stamp*) sea un múltiplo de 32,768 recomienda el uso de tramas F. Como respuesta a la recepción de cualquier tipo de mensaje *media*, excepto las tramas M, se debe enviar un mensaje ACK (Tabla 2.12).

Tabla 2.12. Mensajes media IAX.

Mensaje media	Función
DTMF	Mensaje que lleva solo un dígito DTMF (<i>Dual Tone Multiple Frequency</i>)
Voz	Transporta datos de voz e indica el codec usado
Video	Transporta datos de video e indica el tipo de formato
Texto	Transporta mensajes de texto
Imagen	Transporta una sola imagen
HTML	Transporta HTML y datos relacionados al despliegue de la página
Ruido de comfort	Indica que debe añadirse el ruido de comfort

La Figura 2.17 muestra la secuencia de mensajes utilizados en una llamada de medios IAX:

- *Evento 1:* El par A envía un mensaje NEW al par B para solicitar el inicio de una llamada.
- *Evento 2:* El par B hace una solicitud de autenticación al par A.
- *Evento 3:* En respuesta al mensaje AUTHREQ, el par A envía un mensaje de respuesta de autenticación (AUTHREP).
- *Evento 4:* Se envía un mensaje de aceptación (ACCEPT) al par A.
- *Evento 5:* El par A envía un mensaje ACK como respuesta.
- *Evento 6:* El par B envía una trama F para realizar la configuración de la llamada.
- *Evento 7:* El par A responde con un mensaje (ACK) después de haber recibido la trama F.
- *Evento 8 y 9:* El par B envía tramas M de sonido (ring) al par A.
- *Evento 10:* El par B envía un mensaje RINGING para indicar que se ha procesado el mensaje de petición de llamada y que por lo tanto se tiene que alertar sobre un posible establecimiento de llamada entre los pares.

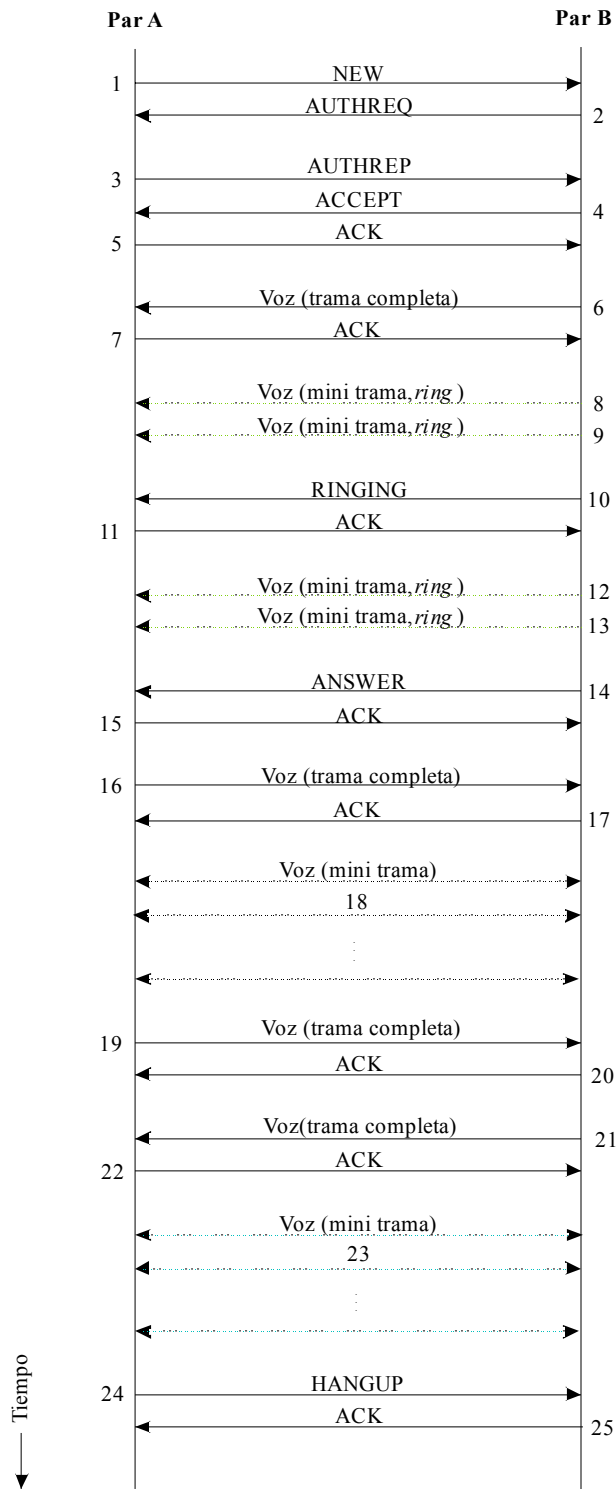


Figura 2.17. Secuencia de mensajes para la transmisión de mensajes *media*.

- *Evento 11*: El par A envía un mensaje ACK indicando que ha recibido la alerta de llamada.
- *Eventos 12 y 13*: El par B continua enviando tramas M de sonido (ring) al par A.
- *Evento 14*: El par B responde a la llamada.
- *Evento 15*: El par A detecta que el par B ha aceptado la llamada y envía un ACK como indicador de que la llamada se ha establecido.

- *Evento 16*: El par A envía una trama F para iniciar el control de la llamada.
- *Evento 17*: El par B responde con un mensaje ACK.
- *Evento 18*: Los pares A y B envían y reciben cualquier tipo de mensajes *media*, en los cuales no se transporta señalización ni control de datos.
- *Evento 19*: El par A envía al par B una trama F para mantener el control de la llamada y evitar desbordamientos.
- *Evento 20*: El par B responde con un ACK para indicar que ha recibido la petición.
- *Evento 21*: El par B envía una trama F al par A para mantener el control de la llamada y evitar desbordamientos.
- *Evento 22*: El par A responde con un ACK indicando que ha recibido la petición del par B.
- *Evento 23*: Los pares A y B envían y reciben cualquier tipo de mensajes *media*, en los cuales no se transporta señalización ni control de datos.
- *Evento 24*: El par A envía un mensaje HANGUP para indicar el termino de la llamada.
- *Evento 25*: El par B acepta el termino de la llamada mediante un mensaje ACK.

La Figura 2.18 muestra la secuencia de mensajes para la transmisión de mensajes *media* vía dispositivos:

- *Evento 1*: El dispositivo IAX envía un mensaje NEW para solicitar iniciar una llamada con el servidor de DialPlan.
- *Evento 2*: El servidor de DialPlan hace una solicitud de autenticación al dispositivo IAX que solicita iniciar una llamada.
- *Evento 3*: Se envía un mensaje de respuesta de autenticación (AUTHREP) por parte del dispositivo IAX.
- *Evento 4*: Se envía un mensaje de aceptación (ACCEPT) al dispositivo IAX en base al mensaje AUTHREP recibido.
- *Evento 5*: El dispositivo IAX envía un mensaje ACK como respuesta.
- *Evento 6*: El dispositivo IAX envía un mensaje DPREQ al servidor de DialPlan para analizar el número de llamada y determinar si hay un patrón de marcado válido.
- *Evento 7*: El servidor de DialPlan envía al dispositivo IAX el dialplan solicitado.
- *Evento 8*: Una vez validado el dialplan, el dispositivo IAX procede a marcar al servidor de DialPlan.
- *Evento 9*: El servidor de DialPlan indica al dispositivo IAX que la llamada está en progreso.
- *Evento 10*: El dispositivo IAX responde con un mensaje ACK de reconocimiento.
- *Evento 11*: El servidor de DialPlan indica al dispositivo IAX que ha contestado a la petición de establecimiento de llamada.
- *Evento 12*: El dispositivo IAX envía un ACK como respuesta al mensaje ANSWER.
- *Evento 13*: El dispositivo IAX envía un trama F de voz al servidor de DialPlan con la finalidad de que el envío sea fiable y se pueda configurar la llamada correctamente.
- *Evento 14*: Se envía un mensaje de reconocimiento (ACK) al dispositivo IAX
- *Evento 15*: El servidor de DialPlan envía un trama F de voz al dispositivo IAX con la finalidad de que el envío sea fiable y se pueda configurar correctamente la llamada previamente establecida.

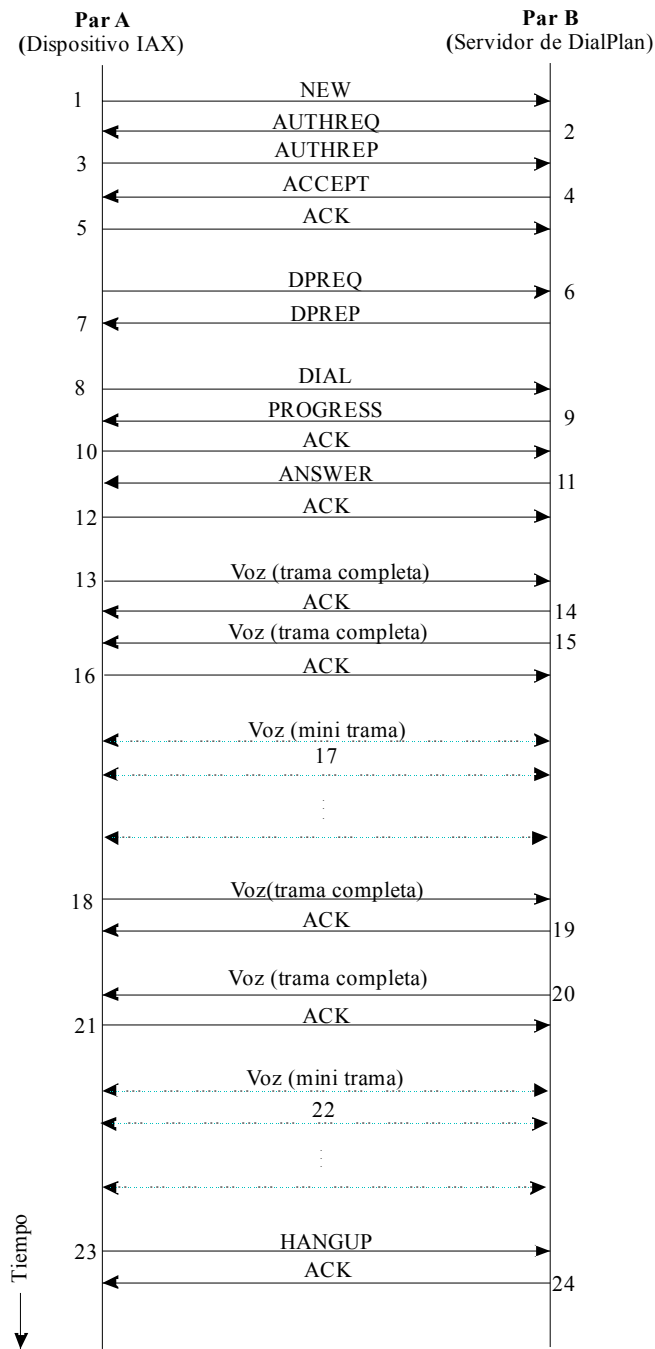


Figura 2.18. Secuencia de mensajes para la transmisión de mensajes *media* vía dispositivos.

- *Evento 16*: Se envía un mensaje de reconocimiento (ACK) al servidor de DialPlan.
- *Evento 17*: Tanto el dispositivo IAX como el servidor de DialPlan envían y reciben cualquier tipo de mensajes *media*, los cuales no transportan señalización ni control de datos.
- *Evento 18*: El dispositivo IAX envía al servidor de DialPlan una trama F de voz con la finalidad de mantener el control de la llamada y evitar cualquier tipo de desbordamiento.
- *Evento 19*: Se envía un mensaje de reconocimiento al dispositivo IAX.
- *Evento 20*: El servidor de DialPlan envía al dispositivo IAX una trama F de voz con la finalidad de mantener el control de la llamada y evitar cualquier tipo de desbordamiento.
- *Evento 21*: Se envía un mensaje de reconocimiento al servidor de DialPlan.

- *Evento 22*: El dispositivo IAX envía el mensaje HANGUP para terminar la llamada.
- *Evento 23*: El servidor de DialPlan envía un mensaje de reconocimiento (ACK) para aceptar el termino de la llamada.

2.4.5. Codificación de mensajes

A continuación se describen las especificaciones para cada tipo de trama definidas por el protocolo IAX.

2.4.5.1. Tramas F

Las tramas F son utilizadas para enviar señalización o datos *media*. Generalmente se utilizan para iniciación de control, configuración y terminación de una llamada IAX; también pueden utilizarse para llevar *stream* de datos aunque no es recomendable. Las tramas F se envían de forma confiable y una vez que se reciben requieren una respuesta de reconocimiento inmediata.

La longitud estándar de la cabecera de una trama completa es de 12 octetos (Figura 2.19), a continuación se describen sus campos:

- *Bit F*: Este bit especifica si la trama es una trama completa (“1” lógico) o no (“0” lógico).
- *Número de llamada origen (source call number)*: Este campo consta de 15 bits y especifica el número de llamada origen.
- *Bit R*: Se utiliza para especificar si la trama está siendo retransmitida o no; si tiene el valor de “0” lógico significa que la trama está siendo retransmitida por primera vez y si contiene el valor de “1” lógico indica que está siendo retransmitida.
- *Número de llamada destino (destination call number)*: Consta de 15 bits y especifica el número de llamada destino, el cual es el mismo que el par remoto tiene como número de llamada origen.
- *Sello de tiempo*: Consta de 32 bits y se utiliza para llevar un control, en milisegundos, desde la primera transmisión de la llamada en curso.
- *Secuencia de salida (OSeqno)*: Campo de 8 bits que controla el número de secuencia del *stream* de salida. Cuando se inicia una llamada su valor es de 0, aumenta cada vez que se envía una trama completa y se reinicializa (*reset*) cuando el contador se desborda.
- *Secuencia de entrada (ISeqno)*: Campo de 8 bits que controla el número de secuencia del *stream* de entrada. Cuando se inicia una llamada su valor es de 0 e incrementa cada vez que se recibe una trama completa y se reinicializa cuando el contador se desborda.
- *Tipo de trama (frametype)*: Identifica el tipo de mensaje que transporta la trama (Apartado 2.4.6).

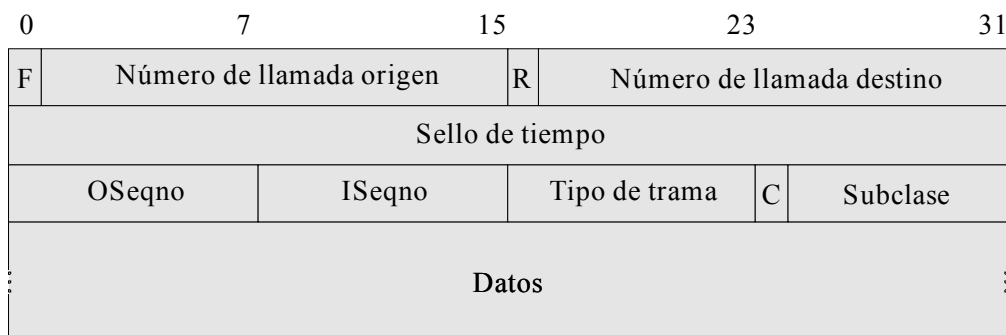


Figura 2.19. Formato binario de una trama F.

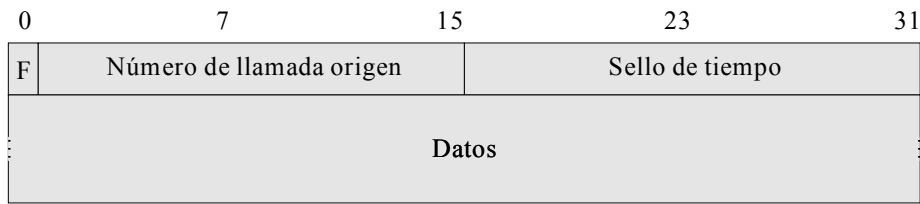


Figura 2.20. Formato binario de una trama M.

- *Bit C*: Determina cuántos de los 7 bits restantes del campo de la subclase se cifran; si el bit C tiene el valor de “1” lógico, el valor de la subclase es interpretado como una potencia de 2; de lo contrario, el valor de la subclase es interpretado como un simple entero sin signo de 7 bits.

2.4.5.2. Tramas M

Son llamadas tramas M debido a que su cabecera consta de tan sólo 4 octetos (Figura 2.20). Las tramas M sólo transportan *media streams* en una llamada ya establecida sin ocuparse del control o señalización de datos.

El control de flujo de las tramas M no es garantizado atendiendo a la especificación de que las llamadas de voz requieren comunicación en tiempo real; sin embargo debe considerarse que generalmente durante una llamada VoIP se pierden tramas, lo cual puede degradar la calidad de la conversación; para asegurar un envío fiable se requeriría la retransmisión de las tramas pérdidas, ocasionando incongruencia en la conversación, lo cual no es permitido en VoIP, ya que las tramas de voz no llegarían en el orden en que fueron enviadas.

Una trama M cuenta con los siguientes campos:

- *Bit F*: Este bit debe tener el valor de “0” lógico para indicar que no es una trama F.
- *Número de llamada origen*: Número utilizado por el par que transmite para identificar la llamada actual.
- *Sello de tiempo*: Las tramas M cuentan con un sello de tiempo de 16 bits y permiten sincronizar las tramas de entrada para procesarlas en orden cronológico, en lugar del orden en que fueron recibidas.

2.4.5.3. Meta tramas

Las meta tramas pueden ser meta tramas de video o meta tramas *trunk*, las cuales se describen a continuación.

2.4.5.3.1. Meta tramas de video

Las meta tramas de video permiten la transmisión de video *streams* con una cabecera optimizada. Son similares en propósito a las tramas M de voz (*mini voice frames*) y cuentan con los siguientes campos (Figura 2.21):

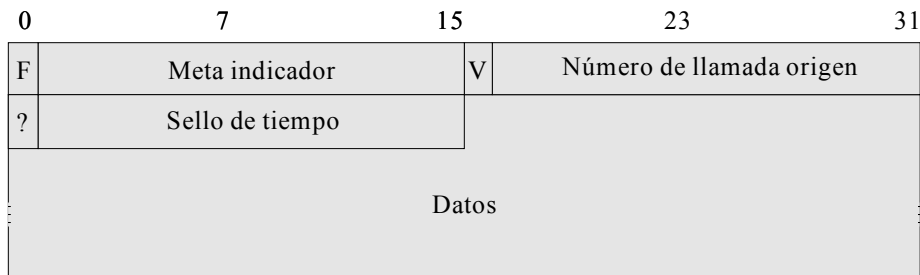


Figura 2.21. Formato binario de una meta trama de video.

- *Bit F*: Un valor de “0” lógico indica que no es una trama F.
- *Meta indicador*: Campo de 15 bits, todos ceros, para indicar que es una meta trama.
- *Bit V*: Tiene el valor de “1” lógico para indicar que es una meta trama de video.
- *Número de llamada origen*: Utilizado por el par que transmite para identificar la video llamada.
- *Sello de tiempo*: Las meta tramas de video cuentan con 16 bits para realizar el sello de tiempo.

2.4.5.3.2. Meta tramas trunk

Las meta tramas *trunk* se utilizan para enlazar múltiples *media streams* IAX entre dos pares en una sola cabecera, con la finalidad de reducir el consumo de ancho de banda. Para ello soporta dos métodos:

- Mediante el envío de una meta cabecera estándar junto a un sello de tiempo de 32 bits, para definir el tiempo de transmisión de la trama *trunk*; después del sello de tiempo se añaden una o más tramas de *media* (*media frames*), las cuales consisten en el número de llamada y la longitud, en octetos, del *stream* de datos.
- Mediante el envío de una meta cabecera estándar junto a un sello de tiempo de 32 bits, para definir el tiempo de transmisión de la trama *trunk*; además de tramas M con un sello de tiempo de 16 bits para cada llamada.

A continuación se describen los campos de las meta tramas *trunk* (Figuras 2.22 y 2.23):

- *Bit F*: Campo de 1 bit que con un “0” lógico indica que no es una trama F.
- *Meta indicador*: Campo de 15 bits, todos ceros, para indicar que es una meta trama.
- *Bit V*: Con un valor de “0” lógico indica que no es una meta trama de video.
- *Meta orden (Meta command)*: Campo de 7 bits para identificar si la meta trama es o no *trunk*. El valor de “1” lógico indica que es una meta trama *trunk* y los demás valores están reservados para uso futuro.

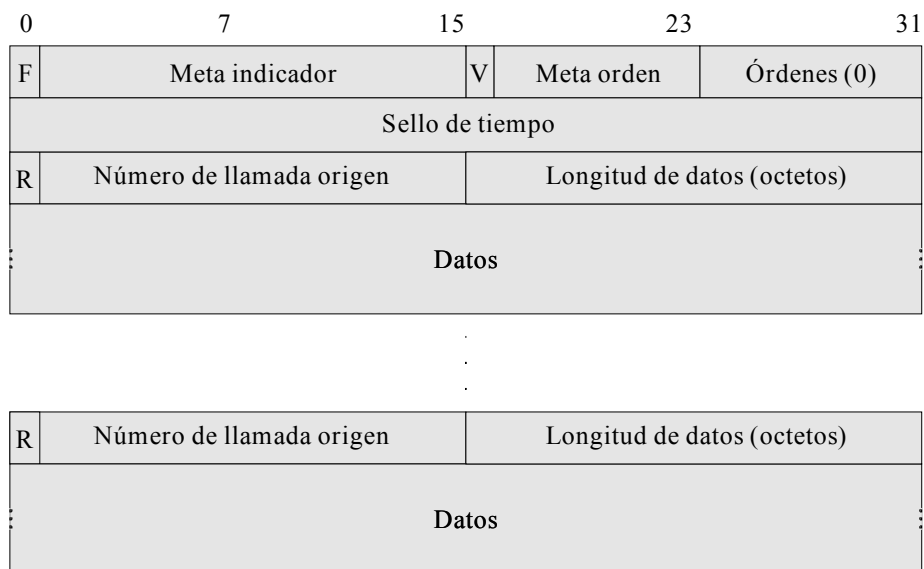


Figura 2.22. Formato binario de la meta trama *trunk* (sello de tiempo = 0).

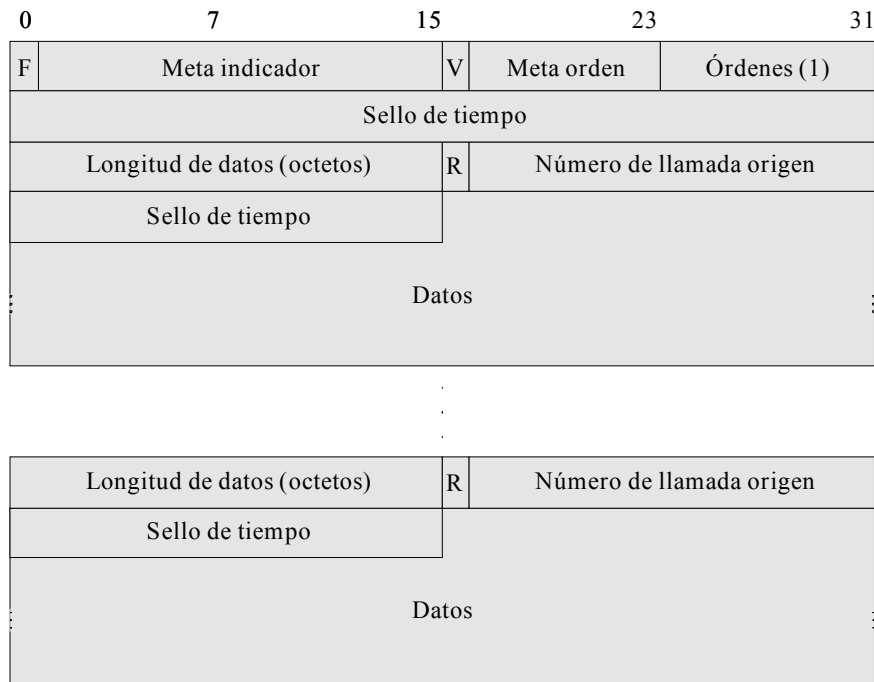


Figura 2.23. Formato binario de la meta trama *trunk* (sello de tiempo = 1).

- *Órdenes (Command data)*: Campo de 8 bits para definir las las opciones que se aplican a una llamada enlazada (*trunked call*); el bit menos significativo del campo es la bandera de los sellos de tiempo *trunk*. Un valor de 0 indica que las llamadas en el enlace no incluyen sus sellos de tiempo individuales. Un valor de 1 indica que cada llamada incluye su propio sello de tiempo.
- *Sello de tiempo*: Las meta tramas *trunk* cuentan con 32 bits para definir el sello de tiempo en el cual se representa el tiempo de transmisión actual de la trama *trunk*. Cabe señalar que dicho sello de tiempo es diferente al definido en las llamadas incluidas en el enlace.

2.4.6. Tipos de trama

El protocolo IAX especifica diez tipos de tramas, lo cual se define en el campo tipo de trama de una trama F (Tabla 2.13).

Tabla 2.13. Tipos de tramas especificadas por el protocolo IAX.

Tipo	Tipo de trama	Función
0x01	DTMF	Transmitir un sólo dígito DTMF
0x02	Voz	Transmitir datos de voz
0x03	Video	Transmitir datos video y especificar el formato
0x04	Control	Transmitir datos del control de la sesión
0x05	Null	Tramas que no deben ser transmitidas
0x06	Control IAX	Transmitir datos de control que proporcionan dirección específica al punto final
0x07	Texto	Transmitir un mensaje de texto no controlado
0x08	Imagen	Transmitir imágenes
0x09	HTML	Transmitir datos HTML
0x0A	Ruido de confort	Transmitir ruido de confort

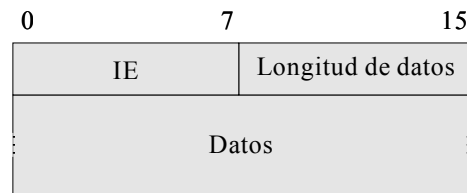


Figura 2.24. Formato binario de un IE.

2.4.7. Elementos de información

Los mensajes IAX enviados como tramas F pueden transmitir IEs para especificar datos del usuario o de una llamada específica. Los IEs se añaden a la cabecera de la trama en el campo de datos. Un mensaje IAX puede incluir uno o múltiples IEs o en su caso no incluir IEs. La Figura 2.24 muestra el formato binario del IE y la Tabla 2.14 describe los IEs definidos por el protocolo IAX.

2.4.8. Encriptado

IAX soporta encriptado de llamadas mediante el uso de llaves simétricas. Utiliza el cifrado por bloques o encriptación avanzada estándar (AES, *Advanced Encryption Standard*). Cuando se envía un mensaje NEW, éste establece que la llamada debe ser cifrada; si la terminal (*host*) destino soporta cifrado de datos, responderá con un mensaje AUTHREQ de texto plano, el cual incluye el IE de ENCRYPTION; todos los mensajes que se envíen posteriormente serán encriptados.

Si la terminal destino no soporta cifrado de datos, el AUTHREQ enviado en respuesta a NEW no incluirá el IE de ENCRYPTION, por lo tanto la terminal origen podrá colgar la llamada o continuarla pero sin cifrar los datos.

2.4.9. Consideraciones de seguridad

IAX soporta tres procesos de autenticación: texto plano (*plaintext*), hash MD5 (*Message-Digest algorithm 5*), y contraseña RSA²⁷ de intercambio. Dichos procesos no consideran el cifrado de medias (*media path*) ni de las cabeceras entre puntos finales, para ello existen soluciones que incluyen el uso de un artefacto de red privada virtual (VPN, *Virtual Private Network*) o de software para encriptar el canal en cualquier otra capa que establezca un método entre los puntos finales con túneles configurados y operacionales.

Existen dos forma de negación de servicio (DoS, *Denial of Service*):

- *Sobrecargando a los pares con peticiones falsas*: Se evita identificando sobrecargas, y emitiendo una alarma o una acción de protección.
- *Ataque ingenioso*: Se puede realizar mediante la inyección de *medias* con la finalidad de ocasionar un exceso de procesamiento al insertar paquetes fuera de orden y enviando órdenes como *hangup* o *transfer*. Estos ataques requieren desactivar la supervisión del canal binario ya que el número de secuencia de los mensajes necesita sincronizarse con el intercambio del protocolo.

Actualmente, se encuentra en investigación el hecho de que IAX pueda cifrar los canales entre los puntos finales mediante el intercambio de llaves RSA o de una llave dinámica en el establecimiento de la llamada, cuya aplicación representa una solución para acoplamiento seguro entre instituciones bancarias, por ejemplo [7].

²⁷ El sistema criptográfico con clave pública RSA es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye (en forma autenticada preferentemente), y otra privada, la cual es guardada en secreto por su propietario.

Tabla 2.14. Elementos de información.

Valor	Nombre	Descripción
0x01	CALLED NUMBER	Número o extensión que se está llamando
0x02	CALLING NUMBER	Número que llama
0x03	CALLING ANI	Identificación automática de número (ANI) que llama para facturación
0x04	CALLING NAME	Nombre del que llama
0x05	CALLED CONTEXT	Contexto para el número
0x06	USERNAME	Nombre de usuario (par o usuario) para autenticarse
0x07	PASSWORD	Contraseña para autenticarse
0x08	CAPABILITY	Capacidad real del codec
0x09	FORMAT	Formato deseado del codec
0x0a	LANGUAGE	Lenguaje deseado
0x0b	VERSION	Versión del protocolo
0x0c	ADSI_CPE	Capacidad CPE ADSI
0x0d	DNID	Originalmente marcado DNID
0x0e	AUTHMETHODS	Método(s) de autenticación
0x0f	CHALLENGE	Censura de datos para <i>MD5/RSA</i>
0x10	MD5 RESULT	Resultado de la censura <i>MD5</i>
0x11	RSA RESULT	Resultado de la censura <i>RSA</i>
0x12	APPARENT ADDR	Dirección visible del par
0x13	REFRESH	Cuando actualizar el registro
0x14	DPSTATUS	Estado del <i>dialplan</i>
0x15	CALLNO	Número de llamada del par
0x16	CAUSE	Causa
0x17	IAX UNKNOWN	Comando IAX desconocido
0x18	MSGCOUNT	Cuántos mensajes esperan
0x19	AUTOANSWER	Solicitud de auto respuesta
0x1a	MUSICONHOLD	Solicitud de música en espera mientras <i>QUELCH</i>
0x1b	TRANSFERID	Solicitud de identificador de transferencia
0x1c	RDNIS	Referencias al servicio de identificación del número marcado (DNIS)
0x1d	PROVISIONING	Información de suministro
0x1e	AESPROVISIONING	Información <i>AES</i> de suministro
0x1f	DATETIME	Fecha/hora
0x20	DEVICETYPE	Tipo de dispositivo
0x21	SERVICEIDENT	Identificador de servicio
0x22	FIRMWAREVER	Revisión del <i>firmware</i>
0x23	FWBLOCKDESC	Descripción del bloque <i>firmware</i>
0x24	FWBLOCKDATA	Bloque de datos del <i>firmware</i>
0x25	PROVVER	Versión de PROVISIONING
0x26	CALLINGPRES	Presentación de la llamada
0x27	CALLINGTON	Tipo de número llamado
0x28	CALLINGTNS	Red de tránsito de la llamada selecta
0x29	SAMPLINGRATE	Velocidades de muestreo soportadas
0x2a	CAUSECODE	Causa por el cual se cuelga la llamada
0x2b	ENCRYPTION	Formato de encriptación
0x2c	ENCKEY	Llave de cifrado AES de 128 bits
0x2d	CODEC_PREFS	Negociación del codec
0x2e	RR JITTER	Retraso entre paquetes recibidos (<i>jitter</i>) recibidos, como en el rfc1889
0x2f	RR LOSS	Perdida recibida, como en el rfc1889
0x30	RR PKTS	Tramas recibidas
0x31	RR DELAY	Retardo máximo del <i>playout</i> para las tramas recibidas en milisegundos.
0x32	RR DROPPED	Tramas caídas (probablemente por el <i>jitterbuffer</i>)
0x33	RR OOO	Tramas recibidas fuera de orden

Los protocolos de establecimiento de llamada más utilizados son H.323, IAX y SIP. H.323 es el protocolo utilizado para videoconferencias y muchos de los dispositivos existentes en el mercado de telecomunicaciones VoIP hacen uso de este protocolo por haber sido el primero para transportar comunicaciones de voz, video y datos; sin embargo su rigidez, difícil implementación y elevado costo está ocasionando que los usuarios estén migrando a SIP por ser un protocolo sencillo con una sintaxis parecida a HTTP y SMTP, además de que los sistemas SIP son de menor costo respecto a los sistemas H.323.

El protocolo IAX cuenta con suficiente información en Internet y es un protocolo par a par orientado a VoIP que presenta transparencia sobre NAT, eficiencia en la asignación del ancho de banda, robustez contra ataques y facilidad de implementación debido a que es código abierto; por otro lado, IAX cuenta con una centralita IP (Servidor / IP PBX) que proporciona las funcionalidades de un conmutador telefónico en hardware y es una solución a bajo costo para transportar cualquier tipo de *media*. Por lo anterior, el presente trabajo de tesis implementara el protocolo de comunicaciones IAX.

3. Desarrollo de la aplicación *softphone XoIP*

Este capítulo describe el desarrollo del *softphone XoIP* con base a los flujos de trabajo del UP.

3.1. Planeación inicial

El *softphone XoIP* se desarrolló empleando el paradigma de la OOP con la finalidad de obtener un software de calidad, durante su ciclo de vida se emplearon las técnicas del UP y el UML para su modelado.

Cabe destacar que en el presente trabajo se utilizó la técnica de reuso composicional en su variante de caja blanca dado que se hizo uso de la librería IAXclient con la finalidad de entender su funcionalidad y adecuarla a las necesidades del proyecto.

3.1.1. Herramientas software involucradas

Las herramientas software utilizadas para desarrollar este trabajo de tesis fueron: Rational Rose Enterprise Edition, QT Designer 3.3 y Asterisk 1.2, bajo los sistemas operativos MS Windows XP y Linux en sus distribuciones Fedora Core 3, Ubuntu 6.06 y Debian Etch.

3.1.1.1. *Rational Rose*

Rational Rose es una herramienta CASE (*Computer Aided Software Engineering*) que apoya a los desarrolladores durante todo el ciclo de vida del desarrollo de un sistema software.

Las herramientas CASE surgieron para dar solución a varios problemas inherentes al diseño del software. Rational Rose hace uso del UML, el cual es considerado como un estándar para el modelado de componentes de un sistema software; Rational Rose está implementado con la finalidad de utilizarse en cualquier medio de aplicación que necesite capturar requerimientos y comportamientos del sistema que se desee construir.

3.1.1.2. *QT Designer*

QT Designer es una herramienta, desarrollada por la firma noruega Trolltech, que permite diseñar, de forma sencilla y rápida, ventanas de diálogo con las librerías Qt [URL2]. Esta herramienta es una aplicación mediante la cual se puede realizar el diseño de una GUI amigable e intuitiva.

QT Designer cuenta con un conjunto de librerías multiplataforma para el desarrollo de GUIs en lenguaje C++ basadas en la OOP. Las librerías Qt se distribuyen bajo licencia libre en dos modalidades: para el público en general (GPL, *General Public License*) o licencia pública (QPL, *QT Public License*), que permiten incorporarlas en aplicaciones de código abierto.

Las librerías Qt se encuentran disponibles para las plataformas: MS Windows, Linux, Mac OS X, Solaris, HP-UX y Unix con X11, además existe una versión para sistemas empujados.

3.1.1.3. Asterisk

Asterisk es una aplicación de código abierto de una central telefónica (PBX) que se encuentra bajo licencia GPL [22]. Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP o a una RDSI.

Inicialmente, Asterisk fue diseñada por Mark Spencer y junto con otros programadores han contribuido a corregir errores y añadir nuevas funcionalidades. Originalmente desarrollado para el sistema operativo GNU/Linux, actualmente también funciona en BSD, MacOSX, Solaris y MS Windows, aunque su plataforma nativa es la que tiene más soporte por parte de la comunidad involucrada en su desarrollo.

Asterisk cuenta con algunas características que sólo estaban disponibles en sistemas propietarios PBX como son: buzón de voz, conferencias, IVR (*Interactive Voice Response*), distribución automática de llamadas, etc. Los usuarios pueden crear nuevas funcionalidades escribiendo un *dial-plan* en el lenguaje *script* de Asterisk o añadiendo módulos escritos en algún lenguaje de programación soportado por Linux [7].

Para conectar teléfonos analógicos se utilizan interfaces telefónicas FXS (*Foreign eXchange Subscriber*)/FXO (*Foreign eXchange Office*), las cuales se encargan de comunicarse con la PSTN.

Asterisk brinda soporte a los protocolos VoIP: SIP, H.323, IAX y MGCP, y puede interoperar con terminales IP actuando como un registrador y como una pasarela. Lejos de competir con las compañías que comercializan soluciones de VoIP de alta calidad como Alcatel, Cisco, Avaya ó Nortel, Asterisk se empieza a incorporar en algunos entornos corporativos como solución de bajo coste junto con SER (*SIP Express Router*).

3.2. Requisitos del *softphone XoIP*

La finalidad de capturar los requisitos es obtener una descripción detallada y clara de lo que el sistema debe realizar, sin olvidar delimitar su alcance (*qué es lo que el sistema debe hacer*).

La captura de requisitos viene a ser una de las partes importantes en el desarrollo de proyectos software durante su ciclo de vida. Respecto a las fases del UP: en la fase de inicio identifica la mayoría de casos de uso para delimitar el sistema y detallar los más importantes; en la fase de elaboración se planifican las tareas que se van a realizar para la elaboración del sistema y se obtiene el resto de los casos de uso; en caso de que se detecten más casos de uso, éstos se capturan e implementan en la fase de construcción; finalmente, en la fase de transición se capturan más requisitos y se modifican los casos de uso si es necesario.

3.2.1. Evaluación de *softphones* existentes en el mercado

Antes de establecer los requisitos necesarios para el análisis, diseño e implementación del *softphone XoIP*, fue conveniente hacer un estudio detallado de los *softphones* existentes tanto en el mercado comercial como en el de distribución libre y gratuita con la finalidad de definir requisitos claros y precisos, delimitando el alcance de los objetivos a lograr en este proyecto de tesis.

Los *softphones* elegidos para el análisis y evaluación fueron seleccionados con base en la demanda de los usuarios de Internet. Son los mismo usuarios los que han determinado qué *softphones* son los más apropiados dependiendo de los codecs y protocolos de establecimiento de llamada. Los *softphones* elegidos para el estudio fueron: Skype [URL13], Xten-Xlite (versión gratuita) [URL20], Idefisk (versión gratuita) [URL7], IaxComm [URL3] y DiAx (versión gratuita) [URL10].

Durante el estudio de los *softphones* mencionados se determinó una serie de criterios, los cuales ayudaron a definir parte de los requisitos del *softphone XoIP*:

- *Codecs soportados*: ¿Cuántos codecs soporta? (subcapítulo 1.5).

- *Protocolo de establecimiento de llamada*: ¿Cuál es el protocolo empleado en el *softphone*? (Capítulo 2).
- *Facilidad de uso*: Considerando la interfaz del usuario ¿Qué tan fácil es adaptarse al ambiente que proporciona el *softphone* para trabajar?
- *Establecimiento de llamadas entre usuarios*: Con base en los estándares existentes ¿Qué tan fácil es responder o realizar una llamada telefónica?
- *Transparencia del IP PBX*: Al momento de poner a funcionar un *softphone* ¿Qué tanto necesita conocer el usuario sobre la configuración del IP PBX?
- *Ayuda proporcionada al usuario*: ¿El *softphone* cuenta con la información suficiente para su uso?

En la Tabla 3.1 se muestran los resultados obtenidos de la comparativa técnica realizada. Cabe mencionar que el estudio realizado respecto a la usabilidad de cada aplicación no fue llevado a cabo mediante un proceso formal de análisis, debido a que no es objetivo de la tesis determinar el nivel de usabilidad de cada *softphone*. En este caso se valoró cada herramienta con base en las existentes.

Tabla 3.1. Comparativa de características técnicas de los *softphones*.

<i>Softphone</i>	Codecs	Protocolo	Usabilidad	Establecimiento de llamada	Transparencia del IP PBX	Ayuda al usuario
Skype	Skype	Skype	Sencillo	Fácil	Totalmente transparente	Suficiente
Xten-Xlite	G.711u, G.711a, GSM, ILBC, Speex	SIP	Sencillo	Fácil	Conocimientos sobre configuración del IP PBX	Suficiente
IdeFisk	G.711u, G.711a, GSM, ILBC, Speex	IAX	Sencillo	Fácil	Transparente y conocimientos IP PBX	Suficiente
IaxComm	G.711u, G.711a, GSM, ILBC, Speex	IAX	Intermedio	Complejo	Conocimientos sobre configuración del IP PBX	No suficiente
DiAx	G.711u, G.711a, GSM, ILBC, Speex	IAX	Sencillo	Fácil	Conocimientos sobre configuración del IP PBX	Suficiente

Se puede resumir que:

- Xten-Xlite es la versión gratuita de la firma counterPath, por tal motivo no cuenta con todas las funcionalidades que tiene el *softphone* comercial de la misma firma.
- Skype es un *softphone* gratuito para llamadas de PC a PC y que cuenta con los servicios de SkypeIn y SkypeOut por los cuales se tiene que pagar una tarifa.
- La versión analizada de IdeFisk es la gratuita, existen otras versiones que son comerciales (BIZ, OEM) aunque fue desarrollado a través de código abierto de la librería IAXclient.
- DiAx es un *softphone freeware* al igual que Xten-Xlite e IdeFisk pero que también fue realizado bajo código abierto del protocolo IAX.
- El único *softphone* de código abierto y gratuito es IaxComm, que cuenta con la mayoría de las funciones de los *softphones* gratuitos anteriores.

Además del resultado mostrado en la Tabla 3.1, fue necesario realizar una comparativa más detallada, en la que se muestra cada una de las funciones de los *softphones* con la finalidad de ver su alcance y elegir las principales características para el desarrollo del *softphone XoIP*. Dicha comparativa de funciones se muestra en la Tabla 3.2, la cual, de forma general, describe la mayoría de funciones que soportan los *softphones*; es importante mencionar que estas funciones pueden ir cambiando conforme salgan nuevas versiones, ya que son *softphones* en constante desarrollo.

Tabla 3.2. Comparativa detallada de *softphones*.

Característica	Skype	Xten-Xlite	IdeFisk	IaxComm	DiAx
DTMF	Sí	Sí	Sí	Sí	Sí
Multilínea	Sí	Sí	Sí	-	-
Transferencia de Llamadas	Sí	Sí	Sí	Sí	Sí
Realizar y recibir llamadas PC a PC	Sí	Sí	Sí	Sí	Sí
Buzón de voz	Sí	Sí	Sí	Sí	Sí
Marcación rápida	Sí	Sí	Sí	-	-
Identificador de llamada	Sí	Sí	Sí	Sí	Sí
Selección y ajuste dispositivos de audio	Sí	Sí	Sí	Sí	Sí
Directorio telefónico	Sí	Sí	Sí	Sí	Sí
Historial de llamadas	Sí	Sí	Sí	-	Sí
Selección del codec	No	Sí	Sí	Sí	Sí
Marcado/remarcado/colgar/responder	Sí	Sí	Sí	Sí	Sí
Conferencia	Sí	Sí	Sí	Sí	Sí
Registro con múltiples servidores	No	Sí	Sí	Sí	Sí
Varios	Puente con la PSTN, codec y protocolo propietario, mensajero instantáneo, panel de marcación, soporte de varios idiomas	Mantiene la línea e ignora llamadas entrantes, temporizador de llamada, <i>threshold</i> , silencio (<i>mute</i>), <i>Push2Talk</i> , selección del sonido, traslado de llamada URI/URL	<i>Pop-up</i> para llamadas entrantes, Función <i>hold</i> (mantener), grabación de llamadas, encriptación del <i>password</i> , tonos de timbre (<i>ring tones</i>) personalizados	<i>Push2Talk</i> , supresión de silencio, función <i>hold</i> (mantener), Carátula (<i>skin</i>), <i>Ring tones</i> personalizados	Soporte completo para los teléfonos (usb ATCOM AU-100/usb YEA-LINK/usb EUTECTICS Inc.), integración Web, soporte de varios idiomas, temporizador de llamada, configuración de latencia de audio, despliegue de número de mensajes viejos y nuevos, autenticación <i>plaintext</i> y MD5, controlado por <i>bluetooth</i> desde un teléfono Sony Ericsson

Una vez analizadas las características técnicas y las funciones de los *softphones*, las más comunes son: soporte de múltiples codecs, IAX es el protocolo más utilizado por desarrolladores de *softphones* de distribución libre y gratuita, utilizan *DTMF*, registro con múltiples servidores, buzón de voz, historial de llamadas, directorio telefónico, temporizador de llamada, identificador de llamada, panel de marcación y selección de dispositivos de audio.

Por tal motivo dichas funciones y características son las que se tomaron como base para el desarrollo del *softphone XoIP*. En cuanto a la usabilidad considerada bajo un proceso no formal de análisis, se determinó que Skype era el que presentaba un menú amigable y sin complicaciones de uso; después le siguió Xten-Xlite, el cual es presentado bajo la forma de un teléfono dejando atrás el concepto de los mensajeros instantáneos; IdeFisk es amigable pero la ubicación de sus elementos funcionales no es de la más adecuada, sin embargo no deja de tener la mayoría de las funciones de los demás *softphones*; DiAx mantiene hasta cierto punto la presentación de IdeFisk, aunque ésta es muy sencilla no deja de ser funcional, además de que combina el uso y configuración de teléfonos IP en hardware; en cuanto a IaxComm, cumple con la mayoría de las funciones de los demás *softphones*, sin embargo su presentación no es la más adecuada.

3.2.2. Actores y casos de uso

Esta actividad es la encargada de delimitar el sistema y su principal objetivo es establecer los actores que interactuarán con el sistema y cuál será su funcionalidad, mediante casos de uso.

De acuerdo al resultado obtenido en el análisis y evaluación de los *softphones*, el *softphone XoIP* tomó como base parte del diseño de la GUI de Skype; en cuanto a funcionalidad se consideraron las características y/o funciones comunes en todos los *softphones* analizados. Por lo tanto el *softphone XoIP* proporciona las siguientes funciones:

- Configurar y crear cuentas de usuario.
- Responder, colgar y realizar llamadas.
- Crear un historial de llamadas entrantes, salientes y perdidas.
- Crear un directorio telefónico de contactos.
- Utilizar un panel de marcación.
- Agregar contactos al directorio telefónico.
- Borrar contactos del directorio telefónico.
- Seleccionar un codec por cada cuenta registrada.
- Seleccionar dispositivos de audio.
- Modificar y eliminar cuentas de usuario.
- Temporizador de llamada.
- Identificador de llamada.

Cabe destacar que existe un sólo actor, el usuario, encargado de manipular la aplicación.

3.2.3. Priorizar casos de uso

Priorizar los casos de uso no es parte del estándar del UML, sin embargo es una buena estrategia que ayuda a establecer prioridades en el análisis y diseño de los mismos [13]; para ello se hace una clasificación de casos de uso primarios, secundarios y opcionales. Los primarios son aquellos que son fundamentales para el funcionamiento de la herramienta, los secundarios aquellos que pueden no ser importantes y los opcionales aquellos que pueden no llevarse a cabo. La Tabla 3.3 muestra la lista de requerimientos obtenidos a partir de las actividades o funciones mencionadas.

Tabla 3.3. Requisitos del *softphone XoIP*.

Primarios	Secundarios	Opcionales
Configurar cuentas	Eliminar cuenta	Conferencia
Configurar cuenta con conexión a servidor	Utilizar agenda telefónica	Log- <i>softphone XoIP</i>
Configurar cuenta con codec preferido	Borrar contacto	
Configurar filtros QoS	Cambiar información del contacto	
Guardar cambios	Agregar contacto	
Crear cuenta	Retener llamada	
Modificar cuenta	Restaurar llamada	
Guardar cuenta	Llamada a buzón de voz	
Iniciar sesión	Ver registro de llamadas	
Cerrar sesión	Llamadas entrantes	
Realizar llamada telefónica	Llamadas salientes	
Recibir llamada telefónica	Llamadas perdidas	
Contestar llamada	Limpiar registro	
Colgar llamada	Recibir llamada adicional	
Cerrar <i>softphone XoIP</i>		

La Figura 3.1 muestra a detalle la relación existente entre los casos de uso.

3.2.4. Función de los casos de uso del *softphone XoIP*

A continuación se describe la función que desempeña cada caso de uso:

- *Configurar cuentas*: Se encarga de obtener los filtros de QoS asignados por el usuario, a su vez se extiende a otros casos de uso: crear cuenta, modificar cuenta y eliminar cuenta.
- *Crear cuenta*: Su función es crear una cuenta con los parámetros necesarios para su correcto funcionamiento.
- *Configurar cuenta con conexión a servidor*: Se encarga de establecer la dirección del servidor, definida por el usuario, al que se conectará la aplicación.
- *Configurar cuenta con codec preferido*: Se encarga de establecer el codificador/descodificador de la voz, asignado por el usuario.
- *Modificar cuenta*: Se encarga de modificar los parámetros de una cuenta previamente creada y configurada.
- *Eliminar cuenta*: Elimina una cuenta a elección del usuario.
- *Guardar cuenta*: Almacena los valores de la cuenta, definidos por el usuario.
- *Configurar filtros QoS*: Se encarga de asignar por parte del usuario cualquiera de los siguientes filtros: reducción de ruido, supresión de silencios, eliminación de eco, enviar tramas de ruido cuando se detecta silencio, detección automática de la ganancia, detección automática de la ganancia analógica, para todas las cuentas creadas.
- *Guardar cambios*: Almacena las opciones de filtro elegidas.
- *Iniciar sesión*: Registra la cuenta elegida por defecto (*default*) en el servidor establecido, de esta manera el usuario puede recibir y realizar llamadas, entre otras funciones.
- *Cerrar sesión*: Desconecta la cuenta por defecto del servidor establecido, evitando realizar y recibir llamadas posteriores.
- *Realizar llamada telefónica*: Proporciona la función de marcar a un número determinado por el usuario, previo inicio de sesión.

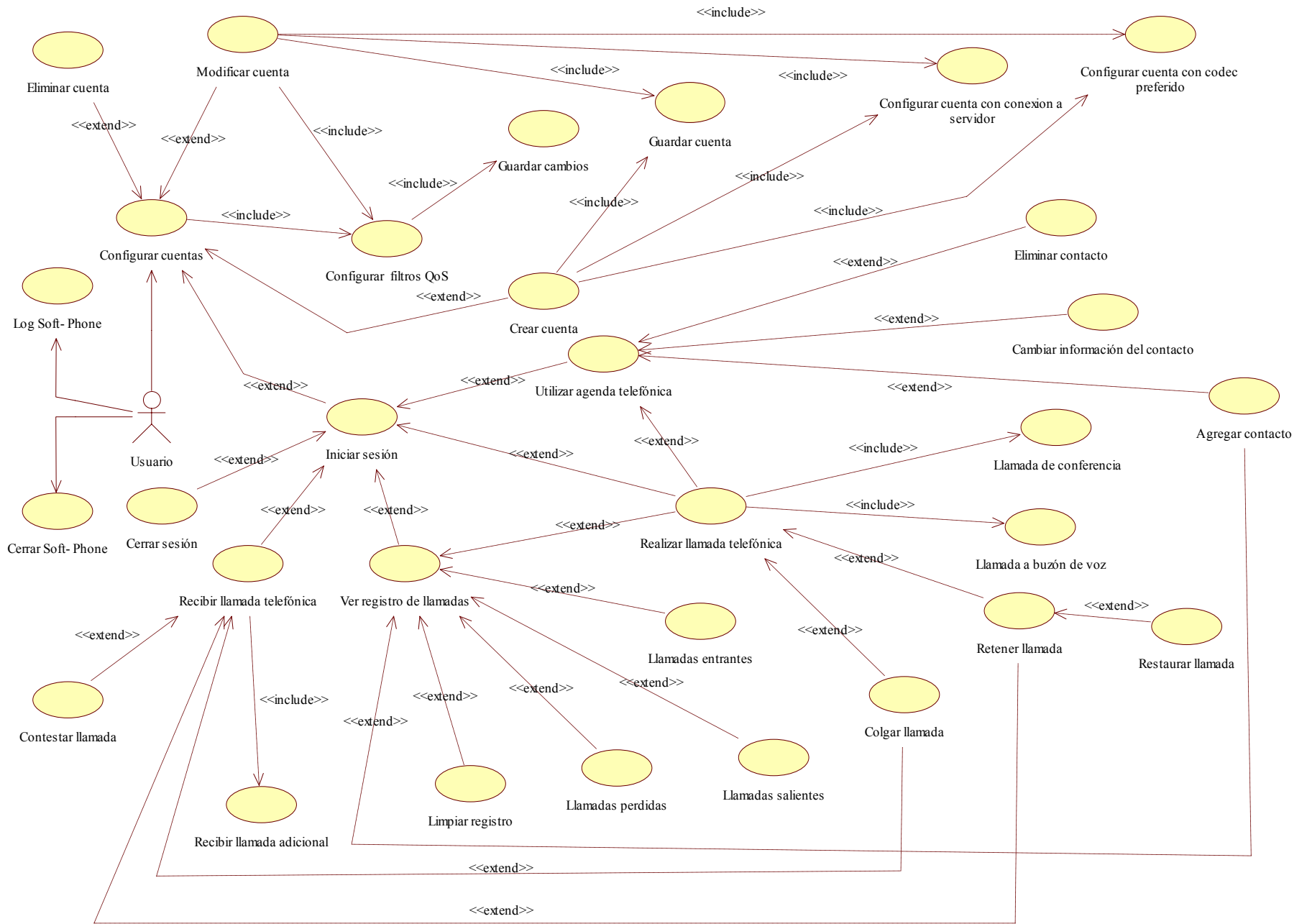


Figura 3.1. Diagrama general de casos de uso del softphone XoIP.

- *Recibir llamada telefónica*: Su finalidad es indicar que la cuenta registrada por el usuario puede recibir llamadas por usuarios registrados en el servidor.
- *Contestar llamada*: Se encarga de responder a llamadas que se reciben por usuarios del mismo servidor.
- *Colgar llamada*: Termina una llamada recibida.
- *Cerrar softphone XoIP*: Cierra la aplicación.
- *Utilizar agenda telefónica*: Permite al usuario acceder a los datos de contactos registrados en su agenda telefónica.
- *Eliminar contacto*: Eliminar un contacto seleccionado de la agenda telefónica.
- *Cambiar información del contacto*: Permite actualizar el número y/o nombre de contacto del usuario seleccionado.
- *Agregar contacto*: Permite al usuario agregar un contacto a la agenda telefónica.
- *Retener llamada*: Retiene una llamada establecida, evitando reproducir los paquetes de voz que se reciban en el punto final (*endpoint*) que tenga activada esta opción.
- *Restaurar llamada*: Restaura una llamada previamente retenida por el usuario, logrando nuevamente volver a reproducir los paquetes de voz.
- *Llamada a buzón de voz*: Permite al usuario marcar a su respectivo buzón de voz, para escuchar los mensajes almacenados.
- *Ver registro de llamadas*: Permite al usuario consultar las llamadas entrantes, salientes y perdidas, así como conocer su identificador de llamada.
- *Llamadas entrantes*: Da a conocer qué personas han llamado y cuántas son las llamadas entrantes recibidas²⁸, así como el tiempo de duración de cada una de ellas..
- *Llamadas salientes*: Da a conocer las llamadas salientes²⁹ y su tiempo de duración.
- *Llamadas pérdidas*: Da a conocer las llamadas que no han sido contestadas por el usuario.
- *Limpiar registro*: Se encarga de eliminar el registro de todas las llamadas de entrada, salida y perdidas.
- *Recibir llamada adicional*: Durante el transcurso de una llamada establecida, permite recibir *n-1* llamadas adicionales que hayan sido especificadas previamente.
- *Llamada de conferencia*: Permite el establecimiento de una llamada telefónica con más de dos usuarios en forma simultánea.
- *Log softphone XoIP*: Enfocado a informar a los desarrolladores de las acciones realizadas durante la ejecución de la aplicación.

3.3. Análisis y diseño

Durante el análisis y diseño del *softphone XoIP* se transformó el modelo de casos de uso en un modelo de diseño con la finalidad de economizar la implementación de los casos de uso, asegurar que el sistema ofrezca un rendimiento adecuado y pueda evolucionar en el futuro [11].

²⁸ Las llamadas entrantes son aquellas que han sido respondidas por el usuario logrando el establecimiento de llamada.

²⁹ Las llamadas salientes son aquellas que han sido marcadas por el usuario y que han sido o no contestadas.

3.3.1. Análisis

En esta etapa se analizan los requisitos, se refinan y se estructuran con la finalidad de comprenderlos y describirlos para facilitar el mantenimiento y construcción de la arquitectura del sistema software. Se puede utilizar un lenguaje formal para señalar detalles relativos a los requisitos del sistema (en UP se conoce como refinar los requisitos), lo cual ayuda a una fácil comprensión, preparación, modificación y en general al mantenimiento de los requisitos. El propósito fundamental del análisis es resolver los requisitos o casos de uso con mayor profundidad, pero a diferencia de la captura de requisitos, se utiliza el lenguaje de los desarrolladores para describir los resultados [11].

En esta etapa se obtienen los siguientes artefactos:

- *Modelos de análisis*: Contienen la realización de los casos de uso y las clases de análisis.
- *Realización de casos de uso*: Es una colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y cómo se ejecuta un determinado caso de uso en términos de las clases y de sus objetos de análisis en interacción.
- *Clases de análisis*: Representan una abstracción de una o varias clases y/o subsistemas del diseño del sistema, definiendo su comportamiento mediante responsabilidades y atributos.
- *Paquete de análisis*: Los paquetes de análisis son utilizados para organizar los artefactos del análisis en piezas manejables.

3.3.1.1. Análisis de la arquitectura

El propósito del análisis de la arquitectura es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis, clases de análisis y requisitos especiales; por lo tanto se deben definir las convenciones de modelado que se adoptarán en las siguientes actividades:

- Todos los paquetes serán identificados con base en los casos de uso definidos.
- Los nombres de las clases principales empezarán con la palabra XoIP y cada una de las palabras que formen el nombre tendrán la primera letra en mayúscula.
- Los nombres de los atributos y operaciones empezarán con minúscula, pero el resto de palabras que conformen el nombre empezarán con mayúscula.

Durante el análisis se lleva a cabo una identificación inicial de los paquetes generales en los que se puede dividir el sistema; éstos se modifican y refinan en la etapa de diseño. Como primera instancia se tiene el modelo de capas típico:

- *Capa de presentación*: Este paquete puede contener todas las clases relacionadas con la interfaz del usuario, por ejemplo: la ventana principal de XoIP y la ventana de configuración de las cuentas de usuario.
- *Capa de reglas de negocio*: Son procesos que implementan e imponen el contexto de los datos dentro del uso especificado de negocios. Es el paquete que contiene todas aquellas clases que controlan las funciones del sistema, por ejemplo la clase que se encarga de comunicar la aplicación con el servidor de registro (Asterisk).
- *Capa de almacenamiento*: Procesos de almacenamiento de datos en el sistema. En este paquete se localizan todas las clases que almacenan la información importante del sistema, por ejemplo la clase que almacenará los valores de las cuentas de usuario y la lista de los contactos.




3.3.1.2. Análisis de los casos de uso

En esta actividad se identifican las clases de análisis asignándoles un estereotipo y se describe el comportamiento de un caso de uso mediante clases de análisis, para lo cual se utilizan los diagra-

mas de colaboración y se realizan los casos de uso.

Una vez identificadas las clases involucradas en el cumplimiento del objetivo de este proyecto de tesis, se procedió a determinar el estereotipo correspondiente a cada una (Tabla 3.4).

Tabla 3.4. Identificación de los estereotipos de las clases

 Boundary	 Control	 Entity
AddUserDialog	laxWrapper	XoIPAddressBook
XoIPContactWindow	laxWrapperEvent	XoIPCallRecord
XoIPAboutBox	XoIPCallRegisterItem	XoIPCallRegister
XoIPAcerca_deUi	XoIPContactListItem	XoIPPrefs
XoIPCallTab	XoIPKeyFilter	XoIPAccount
XoIPCallTabUi	XoIPRegistrationIndicator	XoIPContact
XoIPLogWindowUi	XoIPSystemtray	
XoIPLogWindow		
XoIPMainWindow		
XoIPMainWindowUi		
XoIPPreferenciasUi		
XoIPPrefsWindow		

Las clases tipo *boundary* son aquellas que comúnmente se utilizan para modelar la interacción entre el sistema y sus actores; las clases de tipo *control* son aquellas que representan coordinación, secuencia, transacciones y control de objetos; y las clases tipo *entity* son utilizadas para modelar la información que es duradera y persistente en el sistema, como el almacenamiento de información. Por lo tanto cada uno de estos estereotipos de clase encapsula un tipo diferente de comportamiento (Figura 3.2).

3.3.1.3. Análisis de clases

En esta actividad se procedió a identificar las responsabilidades, atributos, generalizaciones, asociaciones y agregaciones que pudiesen existir en cada una de las clases con la finalidad de determinar un buen análisis de todas las clases involucradas para el desarrollo del *softphone*.

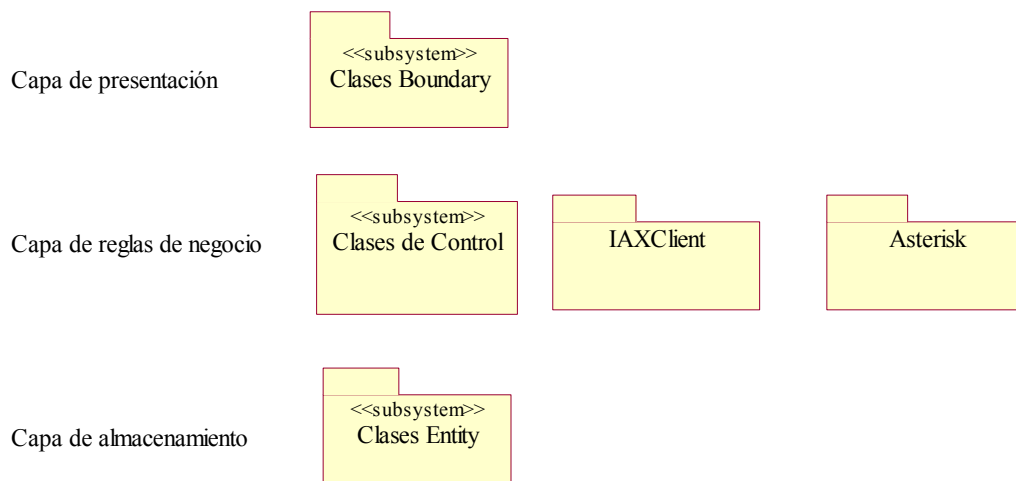


Figura 3.2. Modelo de capas típico de XoIP.

3.3.2. Diseño

En la etapa de diseño se modela el sistema y se da forma, incluida la arquitectura, para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones; para ello los propósitos de esta etapa son [11]:

- Comprender los aspectos relacionados con los requisitos no funcionales y las restricciones respecto a: lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.
- Crear una entrada apropiada y un punto de partida para las actividades de implementación subsiguientes mediante la captura de los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, considerando la posible concurrencia. Lo anterior es útil en los casos en los que no es posible realizar la descomposición basándose en los resultados de la captura de requisitos (incluyendo el modelo de casos de uso) o análisis (incluyendo el modelo de análisis).

Una vez refinado el análisis se obtienen los siguientes artefactos:

- *Modelo de diseño*: Describe la realización física de los casos de uso, los cuales se realizan por las clases de diseño y sus objetos, y se representan mediante realizaciones de casos de uso de diseño.
- *Realización de casos de uso*: Colaboración dentro del modelo de análisis que describe cómo se realiza un caso de uso específico y cómo se ejecuta en términos de la clase de diseño y sus objetos.
- *Clases de diseño*: Son una abstracción similar de una clase o construcción en la implementación del sistema, es decir, el lenguaje utilizado debe ser el mismo que el lenguaje de programación a utilizar. Contiene también la visibilidad de los atributos, la cual puede ser pública, privada o protegida.
- *Paquetes de diseño*: Se encargan de organizar los artefactos del diseño en piezas manejables.

3.3.2.1. Diseño de la arquitectura y los subsistemas

La finalidad de esta actividad es expresar el modelo de diseño y la arquitectura mediante la identificación de subsistemas y sus interfaces, así como las clases de diseño.

En el análisis de la arquitectura se tomó en cuenta una posible clasificación de los paquetes haciendo uso del modelo de capas típico; basándose en dicha clasificación, se consideró que las clases de estereotipo *boundary* pertenecen a la capa de presentación, las de estereotipo *control* pertenecen a la capa de reglas de negocio y las clases de estereotipo *entity* pertenecen a la capa de almacenamiento.

3.3.2.2. Diseño de casos de uso

El diseño de casos de uso permite identificar las clases de diseño y refinar los casos de uso describiendo su comportamiento mediante paquetes, interfaces y clases de diseño.

Con la identificación de las clases de diseño participantes se refinaron las realizaciones de los casos de uso, para ello se estudió cada una de las clases de análisis que participan en la realización de los casos de uso, refinándolas de tal forma que tuvieran una descripción apegada a la implementación. Una vez que se identificaron y realizaron las clases de diseño, se describe cómo interactúan

sus correspondientes objetos de diseño con la finalidad de refinar los diagramas de interacción.

Por cada caso de uso se analizaron los mensajes que se envían entre los objetos de diseño respecto a los diagramas de secuencia y de colaboración de la fase de análisis.

3.3.2.3. *Diseño de clases*

El objetivo de diseñar una clase es cumplir su propósito en las realizaciones de los casos de uso y sus requisitos no funcionales, esto significa que al llegar a esta etapa del modelado se debe contar con clases listas para ser codificadas en un lenguaje de programación.

El diagrama de clases muestra todas las clases que serán utilizadas en la implementación de la aplicación, cada una de las clases cuenta con sus propios atributos y operaciones (véase la información contenida en el CD que acompaña al documento de tesis):

- Los atributos se deben definir utilizando la sintaxis del lenguaje de programación en el que se van a implementar, lo cual incluye especificar el tipo de datos del atributo y la visibilidad de cada atributo.
- Las operaciones, al igual que los atributos, se deben definir y describir utilizando la sintaxis del lenguaje de programación en el que se van a implementar, esto incluye especificar los parámetros con tipo y valores por omisión, el tipo de valor de retorno y la visibilidad de cada operación (*public*, *private* y *protected*).

3.3.2.4. *Identificación de los objetos*

La identificación de los objetos es fundamental en el diseño del sistema debido a que ayuda a definir claramente el objetivo o funcionalidad de cada objeto. A continuación se describen los principales objetos utilizados en los casos de uso (en el CD adjunto se describe la totalidad de ellos):

- *XoIPMainWindow*: Es la clase principal de la aplicación y se encarga de manipular todos los componentes de la GUI.
- *XoIPPrefsWindow*: Clase que se encarga de controlar los parámetros de las cuentas de usuario, tipos de dispositivos de audio, filtros seleccionados por el usuario, cargar los valores de las cuentas cada vez que sean requeridos, seleccionar y obtener el codec deseado, borrar y crear cuentas de usuario y almacenar los cambios de configuración realizados.
- *XoIPPrefs*: Clase utilizada por *XoIPPrefsWindow* para administrar las cuentas de usuario (creación y eliminación de cuentas, control de la cuenta por defecto, carga de los valores de las cuentas, detección de dispositivos de audio y de los filtros disponibles, etc.).
- *XoIPAccount*: Clase encargada de almacenar, eliminar y leer los datos de conexión e identificación de cuentas (Id, alias, nombre, servidor, usuario, clave de acceso, identificador de llamada, número de identificación y codec); también se encarga de indicar el estado de registro de la cuenta (inactiva, conectando, aceptada o rechazada).
- *IaxWrapper*: Clase que hace uso de las funciones de la librería *IAXclient* [URL4].
- *IAXclient*: Librería de código abierto del protocolo IAX [URL4].
- *Servidor Asterisk*: IP PBX utilizado para la administración y control de los pares.
- *XoIPCallRecord*: Clase encargada de llevar el control sobre las llamadas (entrada, salida, duración, nombre y número del usuario).

3.3.3. *Especificación de los casos de uso*

La especificación de los casos de uso describe la totalidad de los casos de uso del *softphone XoIP*, los cuales se pueden consultar en el CD anexo al documento de tesis.

A continuación se describen los casos de uso: crear cuenta, iniciar sesión, realizar llamada telefónica y cerrar sesión, por considerarse los de mayor importancia respecto a la funcionalidad del *softphone XoIP*, cada caso de uso se describe mediante su flujo de eventos y de manera expandida utilizando la tecnología RUP³⁰.

3.3.3.1. Especificación del caso de uso *Crear cuenta*

El caso de uso *Crear cuenta* permite al usuario crear cuentas de usuario, las cuales se encuentran asociadas con un servidor y un codec de voz diferente. El objetivo de este caso de uso es crear una cuenta de usuario en la que se establezca el servidor de conexión y el codec de voz para el establecimiento de llamadas. En este caso de uso se tiene al usuario como único actor.

La Tabla 3.5 muestra el historial de revisiones y la Tabla 3.6 el flujo básico del caso de uso *Crear cuenta*.

Tabla 3.5. Historial de revisiones del caso de uso *Crear cuenta*.

Fecha	Versión	Descripción	Autor
20/Febrero/2006	1.0	Documentación	Alberto Lavariega A.
21/Febrero/2006	1.1	Primeros diagramas	Alberto Lavariega A.
24/Febrero/2006	1.2	Terminación documento	Alberto Lavariega A.
25/Febrero/2006	1.3	Revisión y corrección de errores	Alberto Lavariega A.

Tabla 3.6. Flujo básico del caso de uso *Crear cuenta*.

Actor	Sistema
1. El caso de uso inicia cuando el usuario elige la opción de configuración y dentro de ella selecciona la opción Crear cuenta	2. Muestra un nombre y valores por defecto para la cuenta
3. El usuario escribe el nombre de la cuenta y los datos del servidor al que estará asociada la cuenta	
4. El usuario selecciona el codec de voz deseado para la cuenta	
5. El usuario decide almacenar los valores de la cuenta	6. Verifica que se hayan introducido los parámetros necesarios para la creación de la cuenta
	7. Almacena la cuenta
8. Finaliza el caso de uso <i>Crear cuenta</i>	

Como excepción, si no se introducen los parámetros necesarios para la creación de la cuenta, al momento de almacenar la cuenta, la aplicación devolverá un mensaje indicando que hacen falta parámetros.

Para que el usuario pueda llevar a cabo este caso de uso deberá estar en la ventana de configuración y tener a la mano los datos del servidor y del codec deseado.

Las Figuras 3.3-3.7 muestran los diagramas de actividades, de secuencia, de colaboración, de estado y el diagrama de clases, respectivamente, del caso de uso *Crear cuenta*.

³⁰ Se eligieron los casos de uso de manera expandida por ser aquellos que describen detalladamente un caso de uso de alto nivel, contienen una sección, llamada flujo de eventos, que describe paso a paso la comunicación entre los actores y el sistema, y son útiles para alcanzar un conocimiento más profundo de los procesos y requerimientos establecidos.

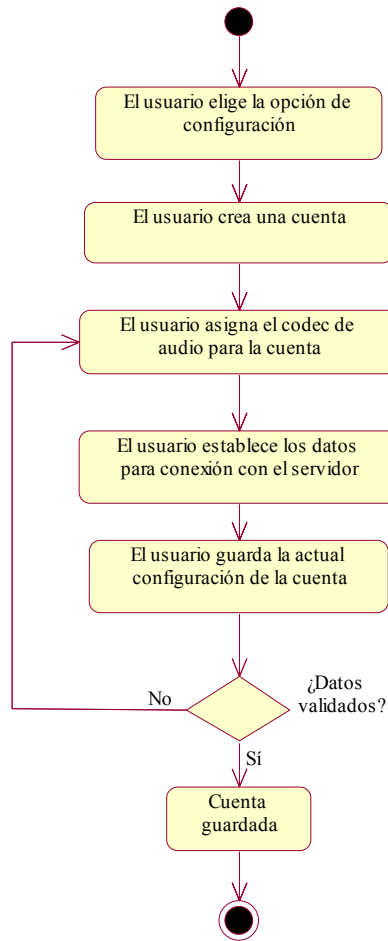


Figura 3.3. Diagrama de actividades del caso de uso Crear cuenta.

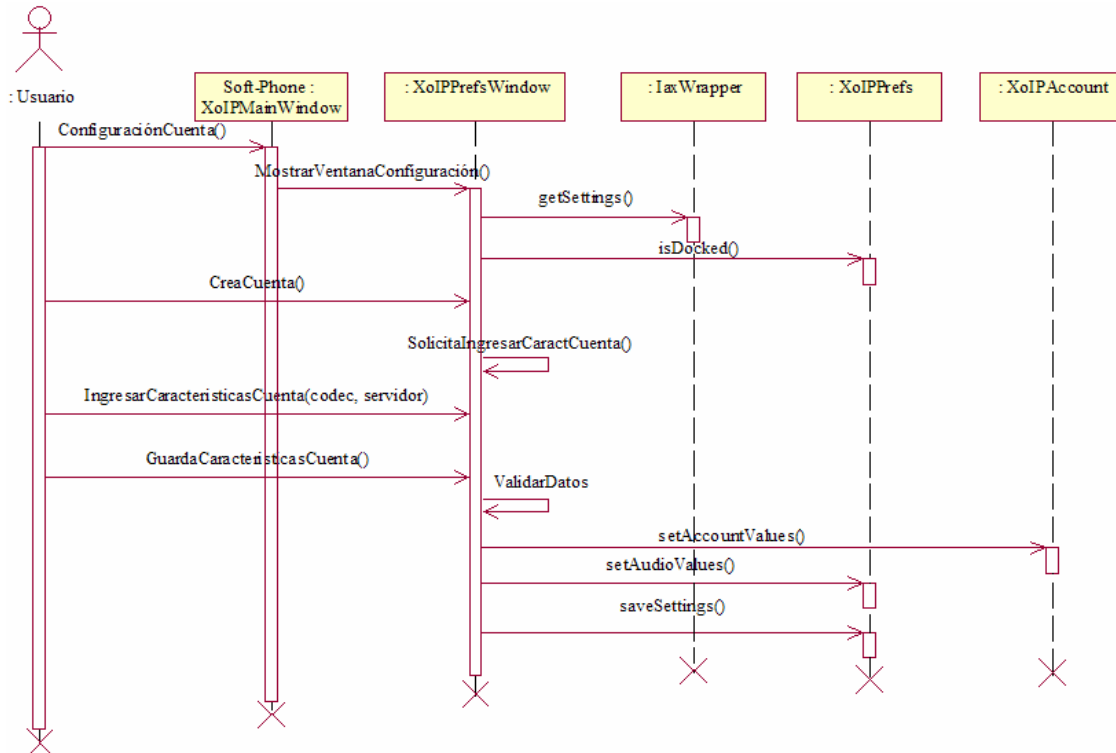


Figura 3.4. Diagrama de secuencia del caso de uso Crear cuenta.

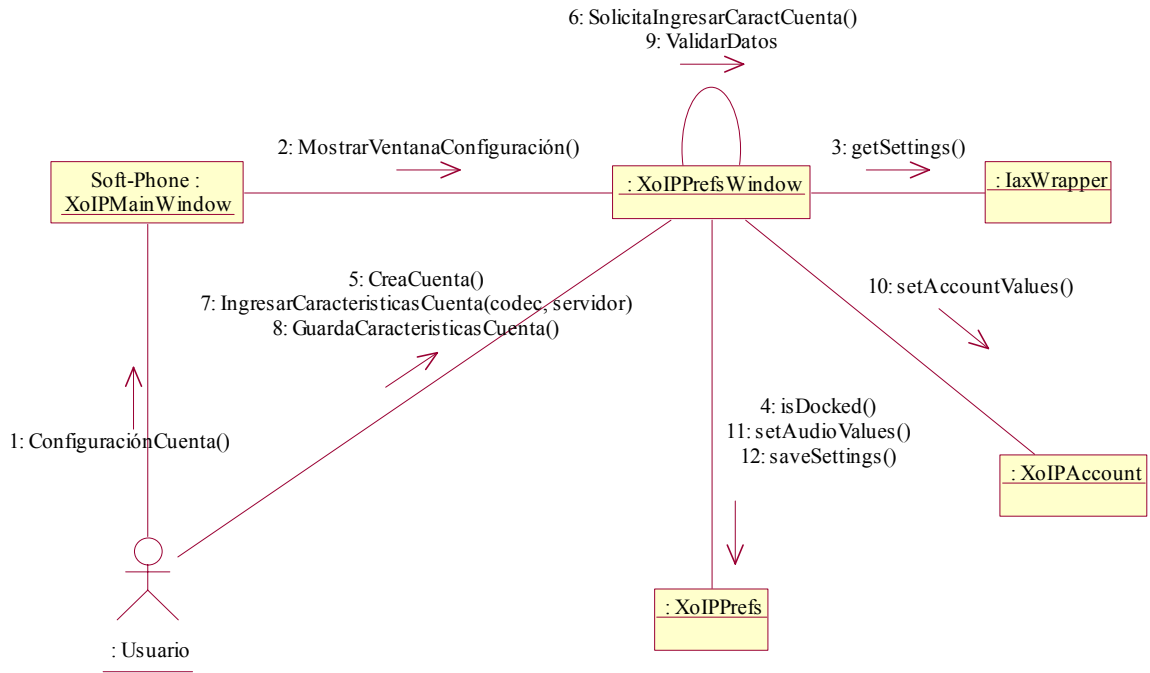


Figura 3.5. Diagrama de colaboración del caso de uso Crear cuenta.

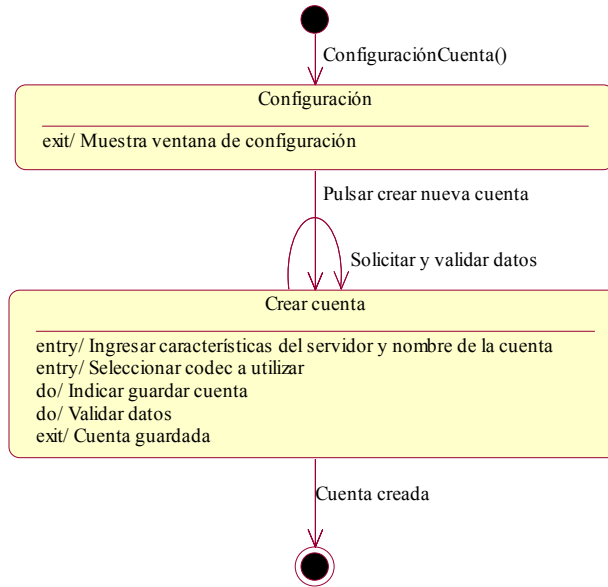


Figura 3.6. Diagrama de estado del caso de uso Crear cuenta.

3.3.3.2. Especificación de caso de uso Iniciar sesión

A través de este caso de uso, el usuario puede iniciar una sesión para poder realizar y recibir llamadas. Cuando se inicia sesión, se establece una conexión con el servidor de la cuenta establecido por defecto. El objetivo de este caso de uso es crear una conexión con un servidor VoIP (IP PBX) con la finalidad de poder realizar y recibir llamadas telefónicas. Este caso de uso tiene un sólo actor, el usuario.

La Tabla 3.7 muestra el historial de revisiones y la Tabla 3.8 el flujo básico del caso de uso Iniciar sesión.

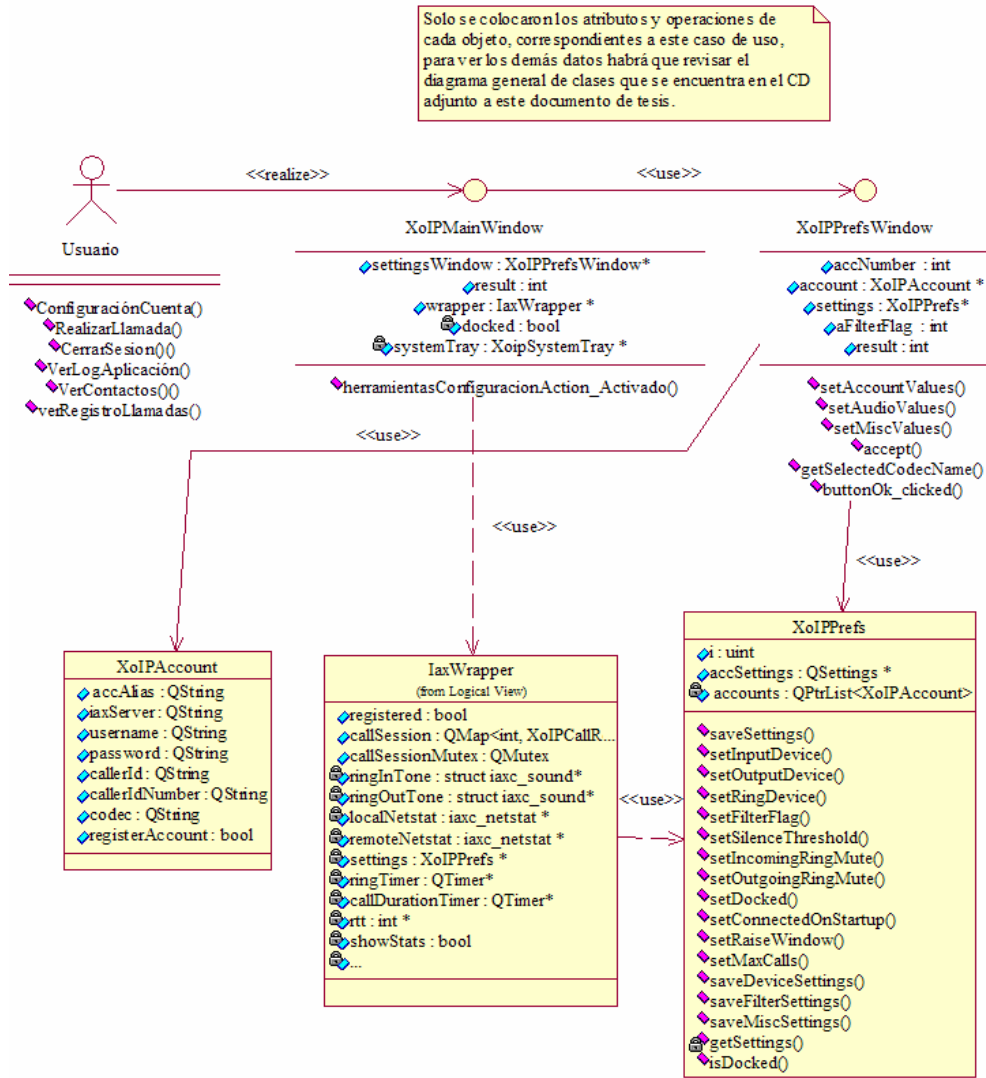


Figura 3.7. Diagrama de clases del caso de uso: Crear cuenta.

Tabla 3.7. Historial de revisiones del caso de uso Iniciar sesión.

Fecha	Versión	Descripción	Autor
26/Febrero/2006	1.0	Documentación	Alberto Lavariega A.
28/Febrero/2006	1.1	Primeros diagramas	Alberto Lavariega A.
01/Marzo/2006	1.2	Terminación documento	Alberto Lavariega A.
02/Marzo/2006	1.3	Revisión y corrección de errores	Alberto Lavariega A.

Tabla 3.8. Flujo básico del caso de uso Iniciar sesión.

Actor	Sistema
1. Este caso de uso inicia cuando el usuario presiona el botón Iniciar sesión	2. El sistema revisa el número de cuentas existentes bajo previa configuración
	3. Revisa si cada cuenta que ha sido configurada previamente cumple con las condiciones para ser registrada
	4. Registra cuentas
	5. Indica el estado de registro establecido con el servidor de la cuenta por defecto
6. Finaliza el caso de uso	

Como excepción se tiene que si no se cumplen las condiciones o parámetros para que una cuenta pueda ser registrada, el servidor devuelve un mensaje de rechazo.

Como precondition el usuario debe haber configurado previamente la(s) cuenta(s) a ser registrada(s).

Las Figuras 3.8-3.12 muestran los diagramas de actividades, de secuencia, de colaboración, de estado y el diagrama de clases, respectivamente, del caso de uso Iniciar sesión.

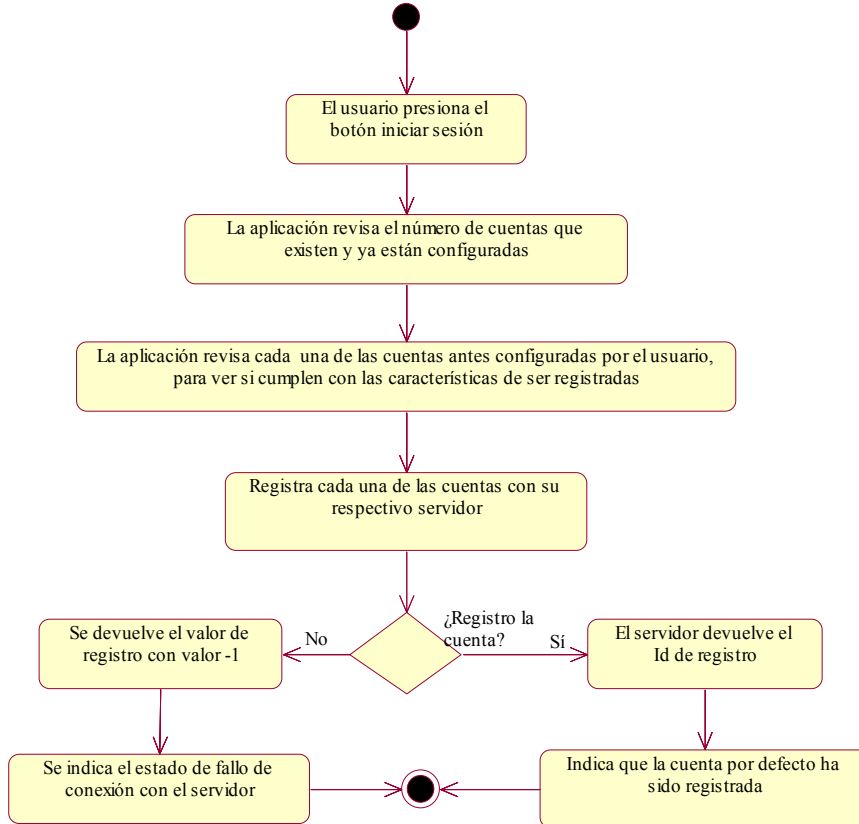


Figura 3.8. Diagrama de actividades del caso de uso Iniciar sesión.

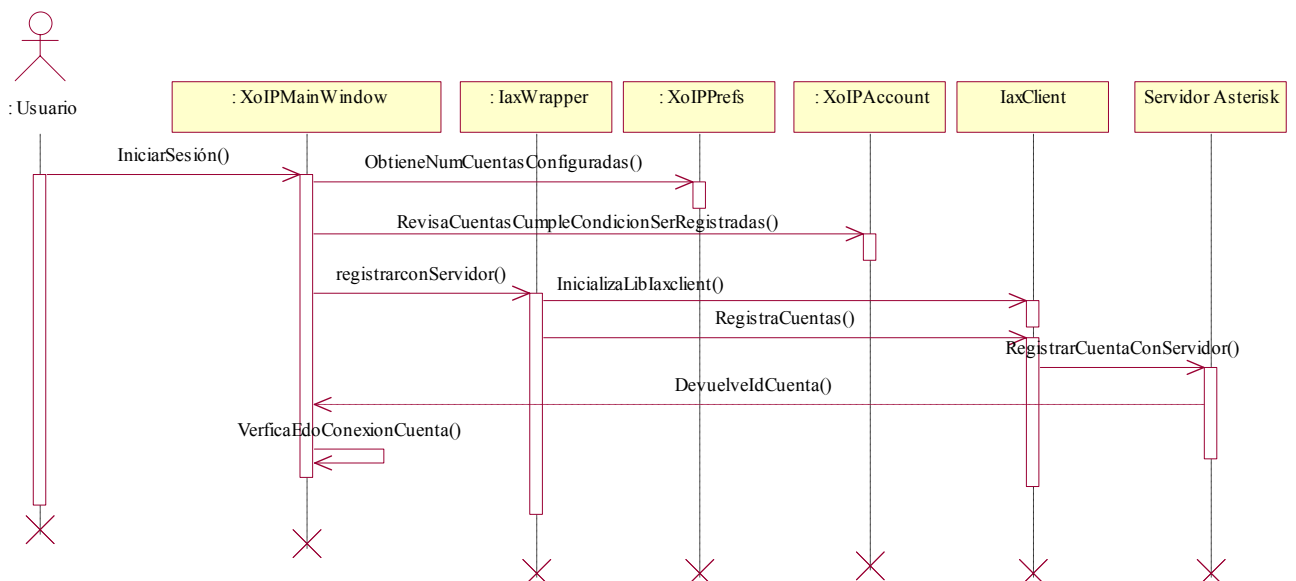


Figura 3.9. Diagrama de secuencia del caso de uso Iniciar sesión.

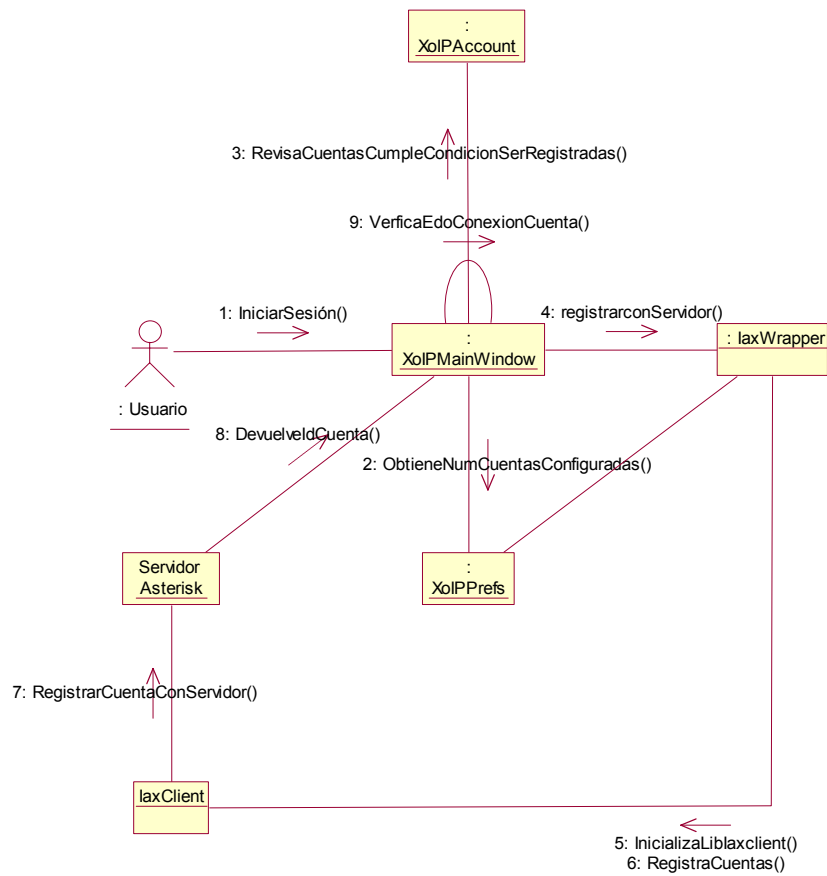


Figura 3.10. Diagrama de colaboración del caso de uso Iniciar sesión.

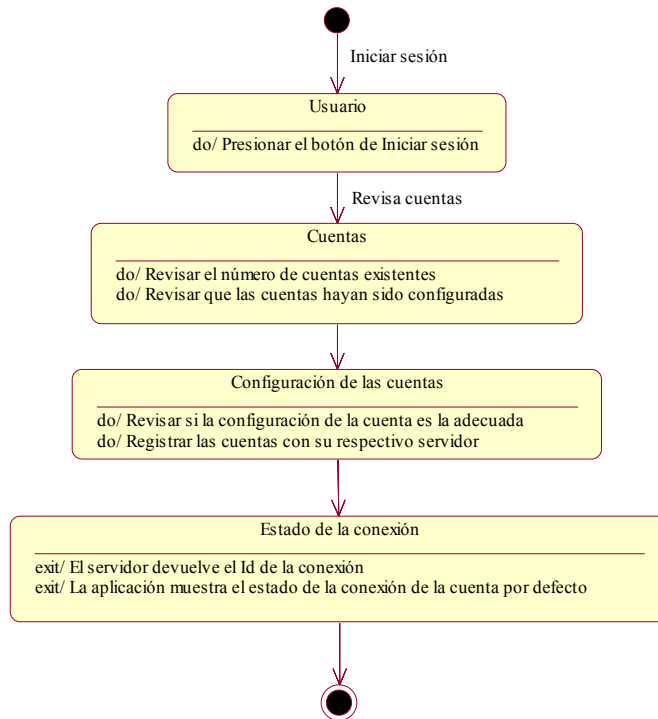


Figura 3.11. Diagrama de estado del caso de uso Iniciar sesión.

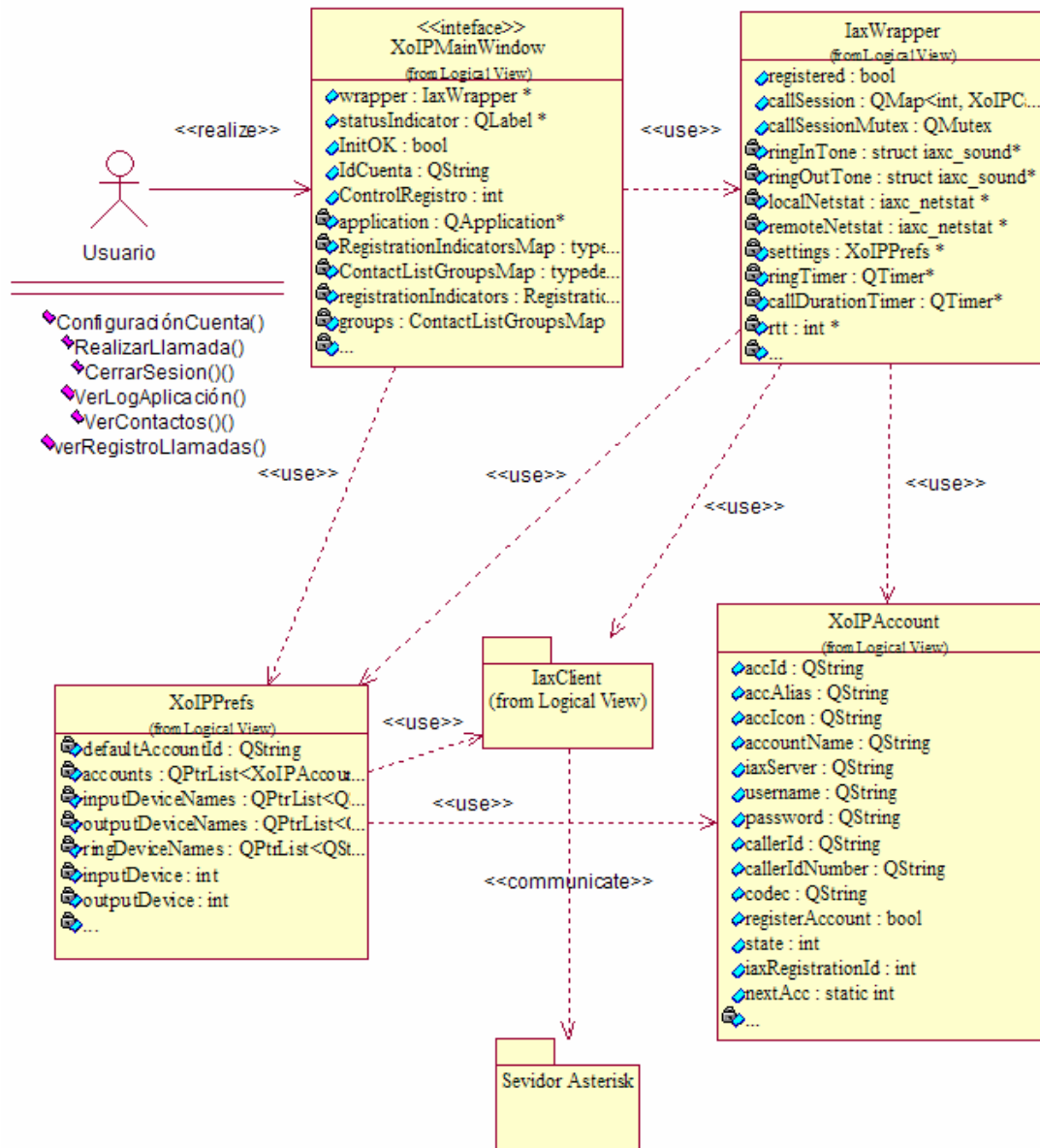


Figura 3.12. Diagrama de clases del caso de uso Iniciar sesión.

3.3.3.3. Especificación de caso de uso Realizar llamada telefónica

Con este caso de uso, el usuario puede realizar llamadas telefónicas IP de PC a PC, PC a teléfono IP o análogo, con usuarios del mismo o diferente IP PBX, o haciendo uso de la PSTN siempre y cuando se cuente con una pasarela. El objetivo de este caso de uso es poder realizar llamadas telefónicas IP entre usuarios registrados en el IP PBX. La Tabla 3.9 muestra el historial de revisiones y la Tabla 3.10 el flujo básico del caso de uso Realizar llamada telefónica.

Tabla 3.9. Historial de revisiones del caso de uso Realizar llamada telefónica.

Fecha	Versión	Descripción	Autor
03/Marzo/2006	1.0	Documentación	Alberto Lavariega A.
04/Marzo/2006	1.1	Primeros diagramas	Alberto Lavariega A.
05/Marzo/2006	1.2	Terminación documento	Alberto Lavariega A.
06/Marzo/2006	1.3	Revisión y corrección de errores	Alberto Lavariega A.

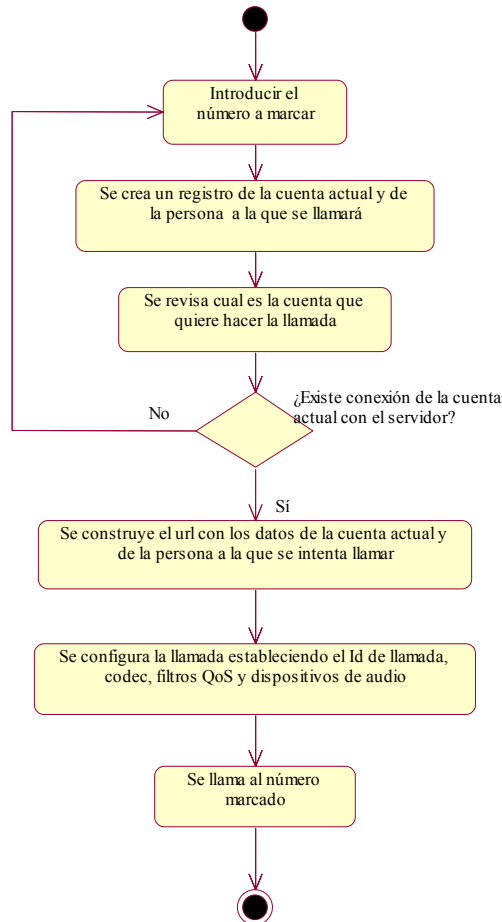
Tabla 3.10. Flujo básico del caso de uso Realizar llamada telefónica.

Actor	Sistema
1. El usuario introduce el número a marcar o selecciona el contacto de la agenda telefónica al que invitará a establecer una conversación telefónica por IP	2. Crea un registro de la cuenta actual y de la persona a la que llamará
	3. Toma a la cuenta por defecto como la cuenta que quiere establecer una conversación telefónica
	4. Revisa que exista una conexión de la cuenta actual con la IP PBX
	5. Construye el URL de llamada
	6. Configura la llamada
	7. Llama al número o contacto indicado por el usuario
	8. Conversación telefónica IP en curso
9. Termina el caso de uso	

Como excepción se considera que si el contacto o número al que se llama no se encuentra registrado, no establecerse la llamada.

La precondition de este caso de uso es que el usuario deberá haber iniciado sesión con la cuenta establecida por defecto.

Las Figuras 3.13-3.17 muestran los diagramas de actividades, de secuencia, de colaboración, de estado y el diagrama de clases, respectivamente, del caso de uso Realizar llamada telefónica.

**Figura 3.13.** Diagrama de actividades del caso de uso Realizar llamada telefónica.

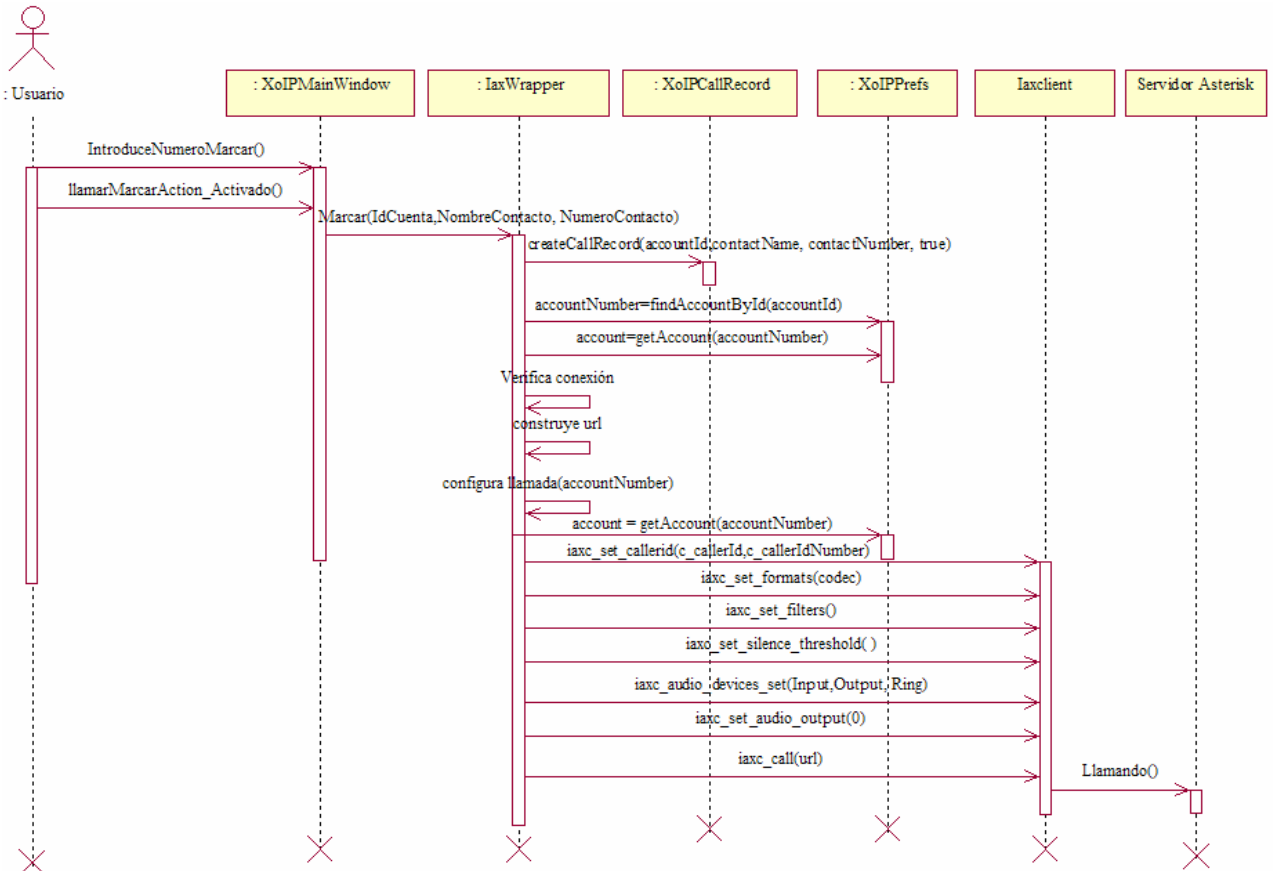


Figura 3.14. Diagrama de secuencia del caso de uso Realizar llamada telefónica.

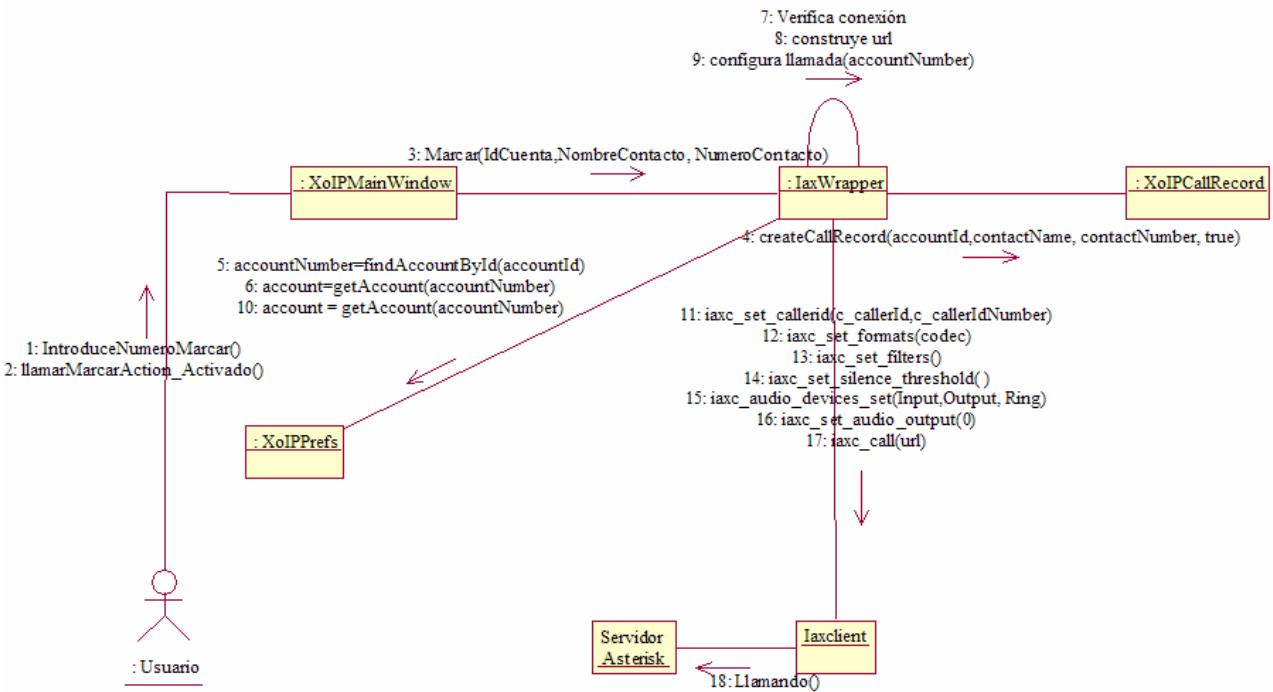


Figura 3.15. Diagrama de colaboración del caso de uso Realizar llamada telefónica.

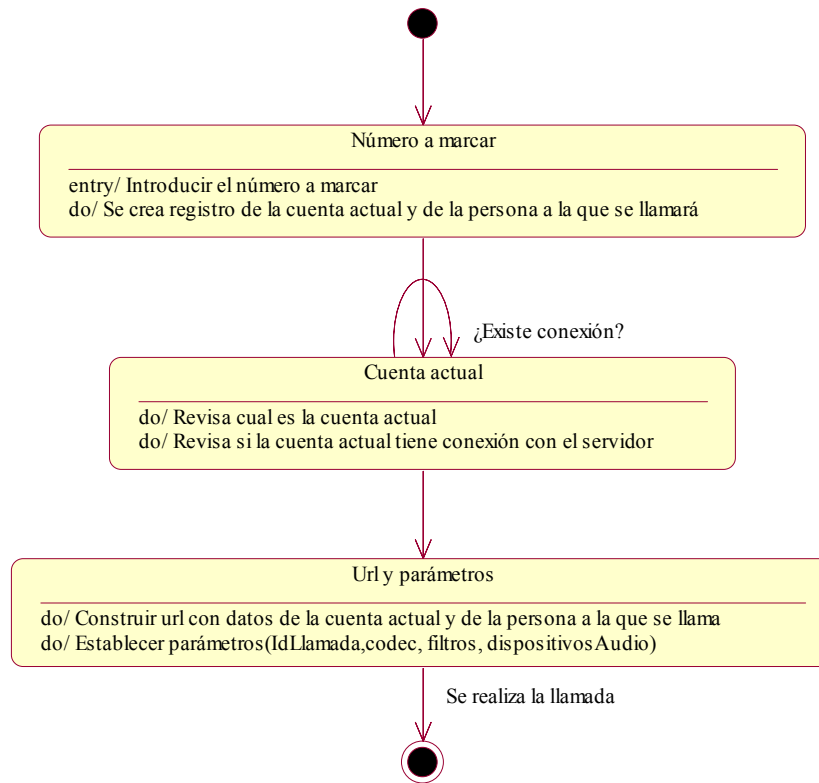


Figura 3.16. Diagrama de estado del caso de uso Realizar llamada telefónica.

3.3.3.4. Especificación de caso de uso Cerrar sesión

A través de este caso de uso, el usuario puede desconectar la aplicación del servidor VoIP, inhabilitando el establecimiento y recepción de llamadas. El objetivo es que el usuario pueda cerrar la sesión. Este caso de uso tiene un sólo actor, el usuario. La Tabla 3.11 muestra el historial de revisiones y la Tabla 3.12 el flujo básico del caso de uso Cerrar sesión.

Tabla 3.11. Historial de revisiones del caso de uso Cerrar sesión.

Fecha	Versión	Descripción	Autor
07/Marzo/2006	1.0	Documentación	Alberto Lavariega A.
08/Marzo/2006	1.1	Primeros diagramas	Alberto Lavariega A.
09/Marzo/2006	1.2	Terminación documento	Alberto Lavariega A.
10/Marzo/2006	1.3	Revisión y corrección de errores	Alberto Lavariega A.

Tabla 3.12. Flujo básico del caso de uso Cerrar Sesión.

Actor	Sistema
1. Este caso de uso comienza cuando el usuario presiona el botón Cerrar sesión	2. Obtiene la(s) cuenta(s) que habrá(n) de desconectarse del servidor VoIP
	3. Desconecta la(s) cuenta(s) del servidor
	4. Indica que la(s) cuenta(s) ha(n) sido desconectada(s)
	5. Detiene la ejecución de la librería IAXClient
	6. Muestra el mensaje de iniciar sesión debido a que la aplicación ha sido desconectada del servidor
7. Finaliza el caso de uso	

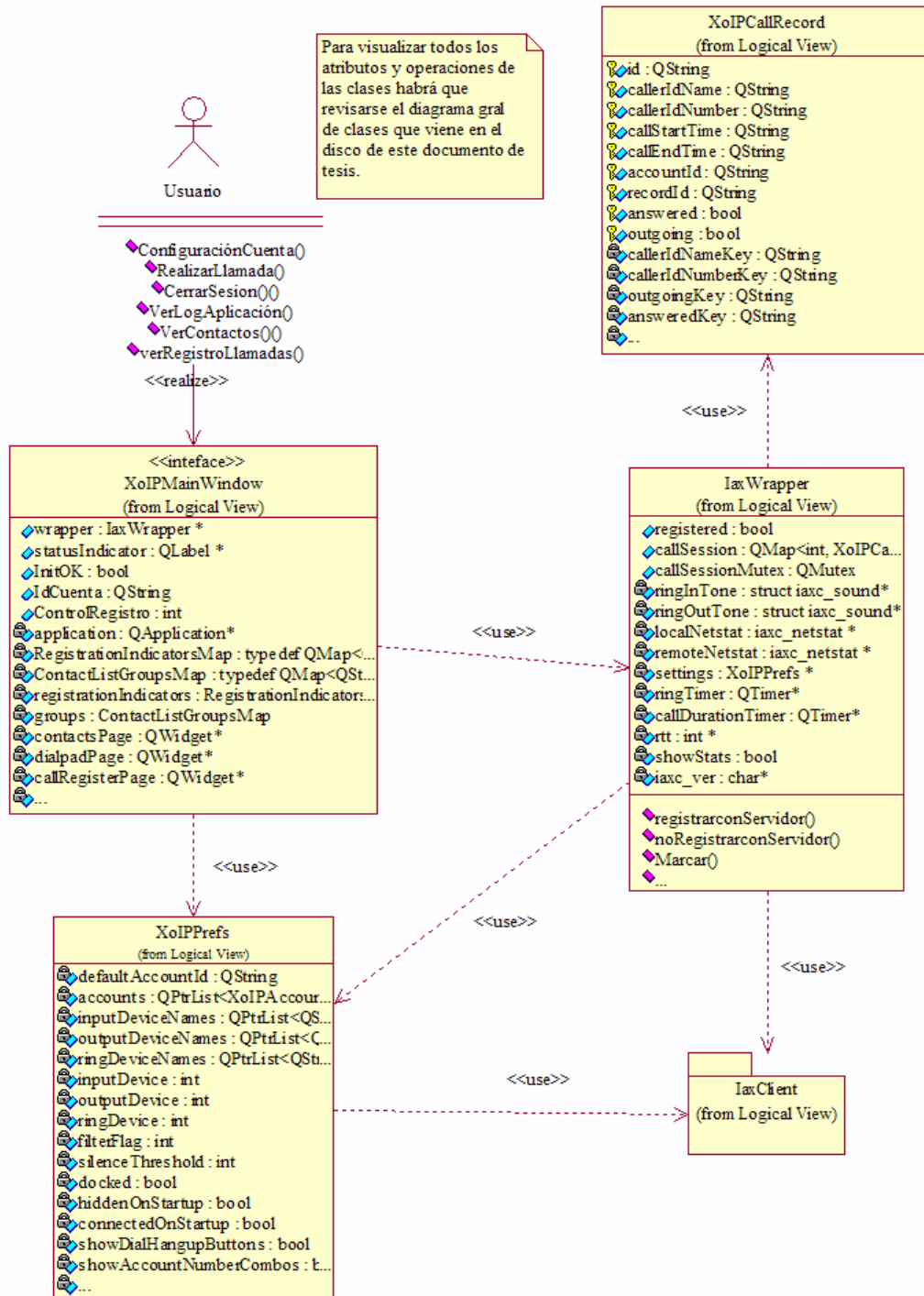


Figura 3.17. Diagrama de clases del caso de uso Realizar llamada telefónica.

Como excepción se considera que si al momento de cerrar la sesión no existe conexión al servidor, la aplicación se detiene como si la hubiera. Para cuando ya exista red, la aplicación ya no estará revisando su conexión con el servidor puesto que ya no existe.

La precondition de este caso de uso es que el usuario debió haber iniciado sesión previamente.

Las Figuras 3.18-3.22 muestran los diagramas de actividades, de secuencia, de colaboración, de estado y el diagrama de clases, respectivamente, del caso de uso Cerrar sesión.

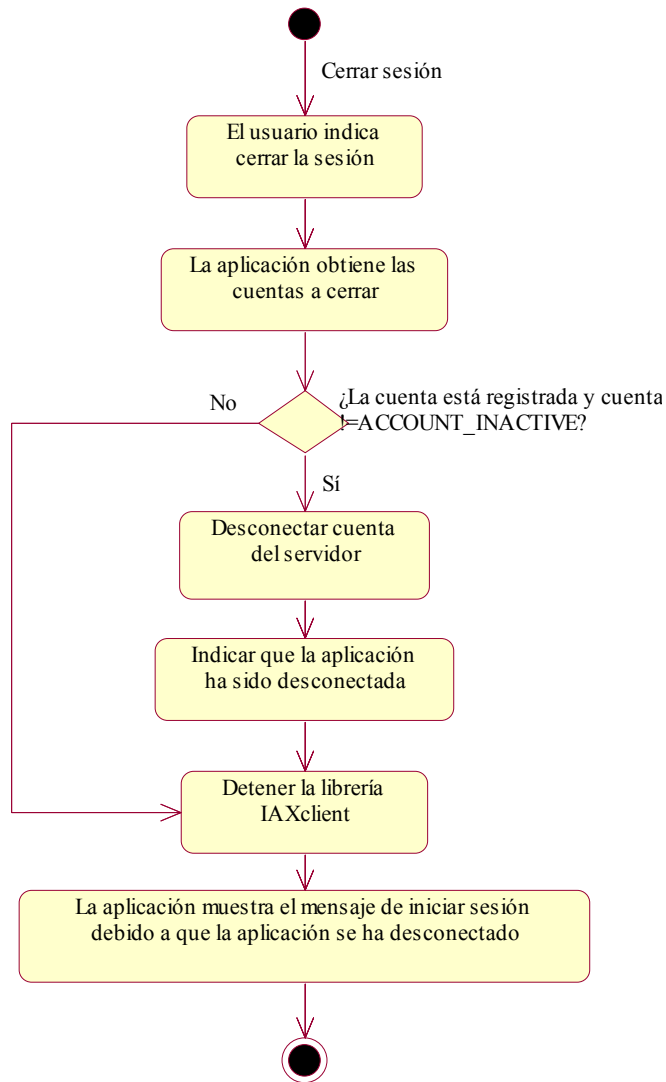


Figura 3.18. Diagrama de actividades del caso de uso Cerrar sesión.

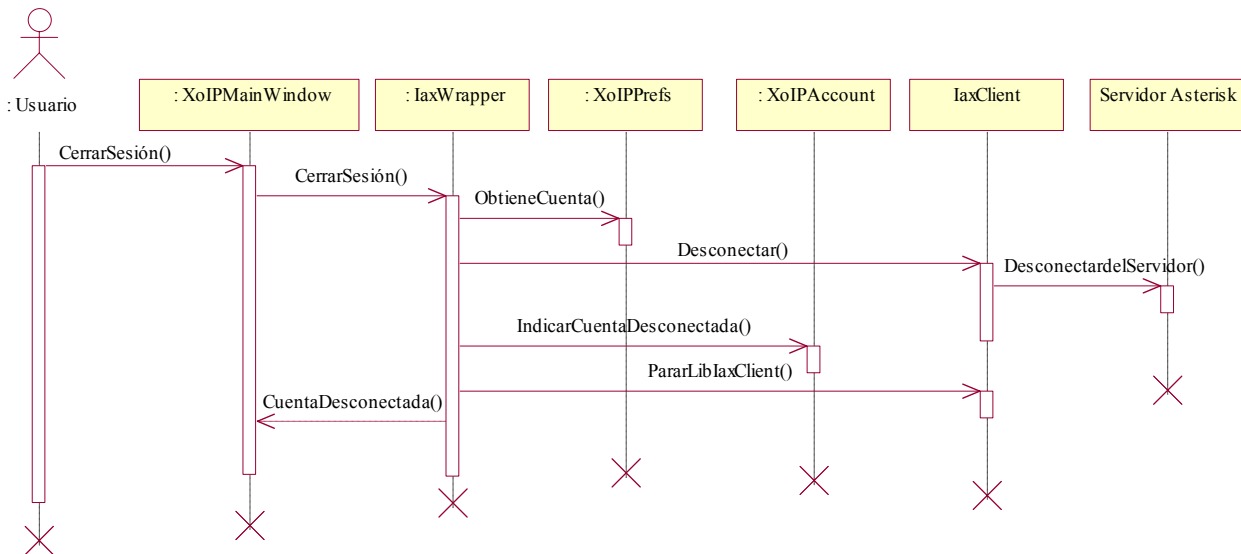


Figura 3.19. Diagrama de secuencia del caso de uso Cerrar sesión.

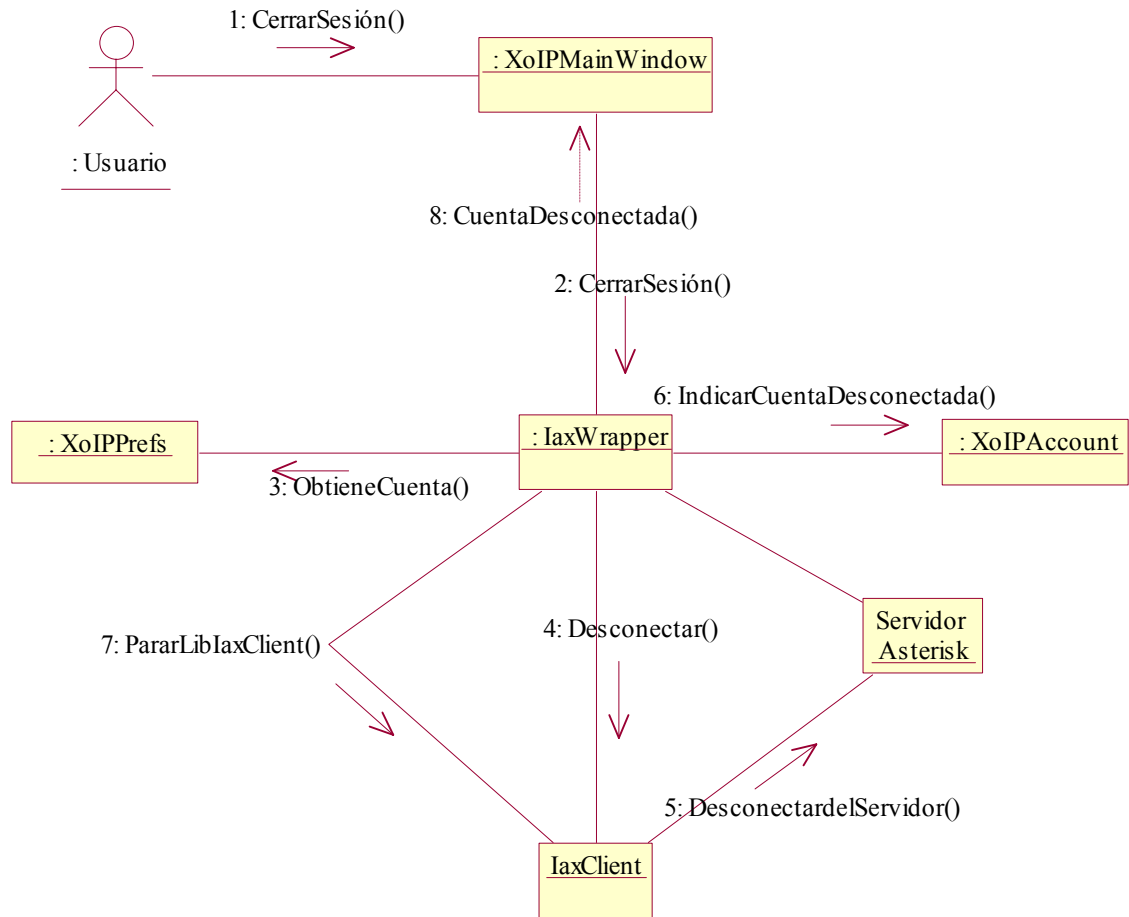


Figura 3.20. Diagrama de colaboración del caso de uso Cerrar sesión.

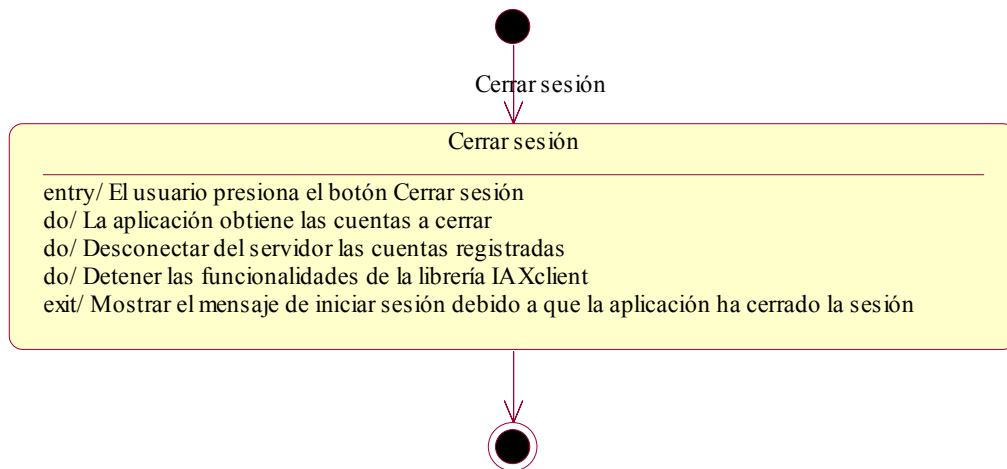


Figura 3.21. Diagrama de estado del caso de uso Cerrar sesión.

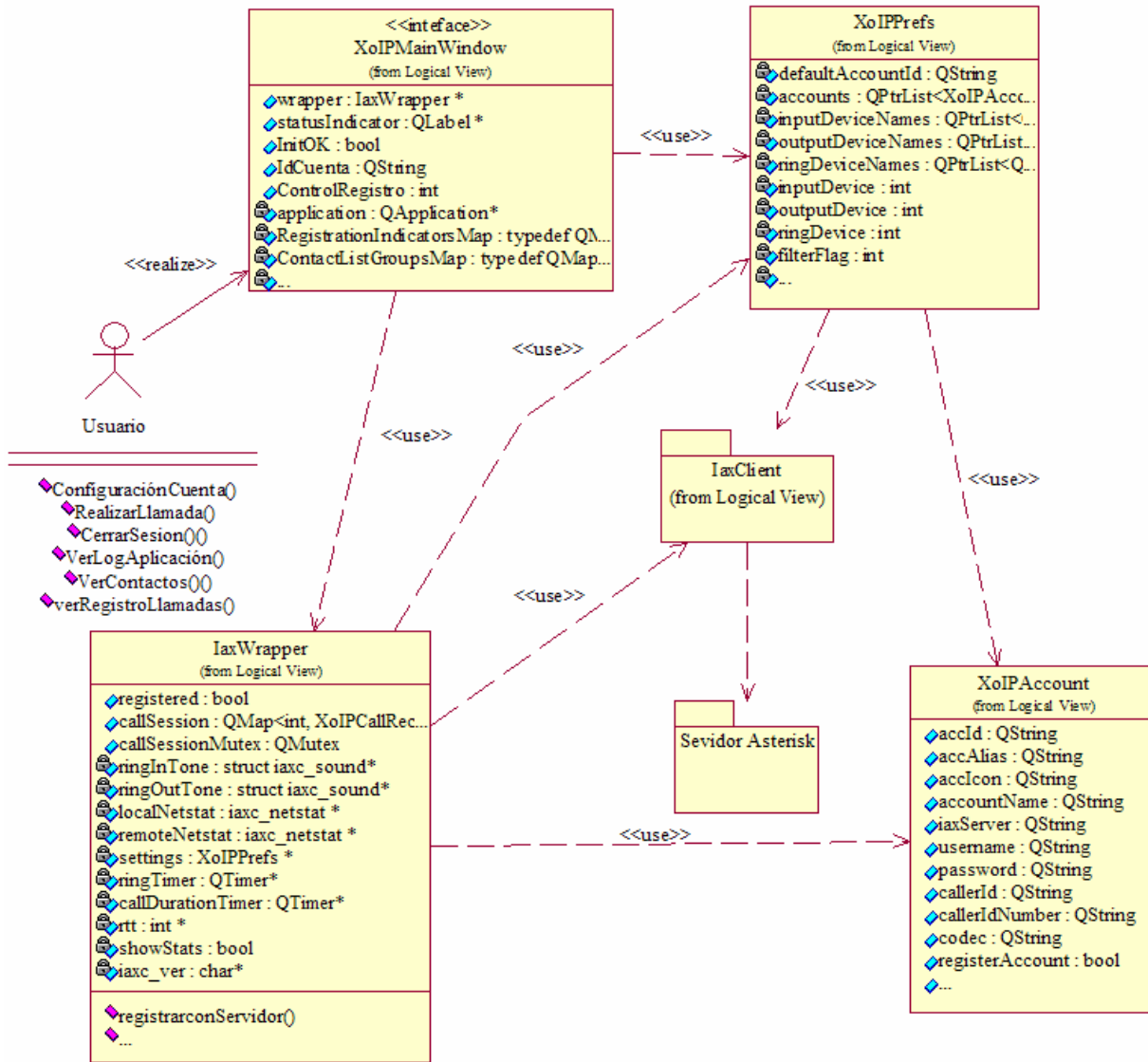


Figura 3.22. Diagrama de clases del caso de uso Cerrar sesión.

3.4. Implementación

En este flujo de trabajo se tomó el resultado del diseño del *softphone XoIP* para implementarlo en términos de componentes, es decir, ficheros de código fuente, binario, *scripts*, ejecutables y similares.

Los artefactos que se obtuvieron en este flujo de trabajo fueron los siguientes:

- *Modelo de implementación*: Es la implementación de los elementos del diseño en componentes y una descripción de la organización de dichos componentes de acuerdo al entorno de implementación y al lenguaje de programación utilizado.
- *Componente*: Es la parte física de los elementos del diseño (clases, paquetes, etc.).
- *Subsistema de implementación*: Es la evolución de los paquetes del diseño, organiza los componentes de implementación en piezas manejables. Se representa por medio de un mecanismo de empaquetamiento que depende del ambiente de programación, por ejemplo: un paquete en java, un directorio de archivos en C++ o bien un proyecto en el caso de QT Designer.

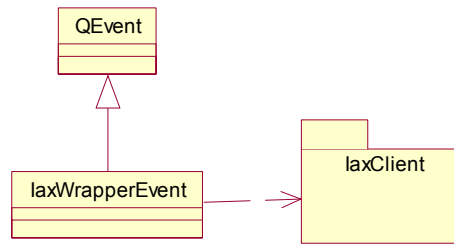


Figura 3.23. Ejemplo de reuso de la clase QEvent de QT Designer y de la librería IAXclient.

3.4.1. Reuso en la implementación

Lo primero que se hizo en este flujo de trabajo fue identificar los componentes más importantes y significativos del *softphone XoIP* para proceder con su respectiva codificación. Basándose en la priorización de casos de uso, los primeros casos de uso en ser implementados fueron los *primarios*, posteriormente los *secundarios* y por último los *opcionales*.

Durante la etapa de codificación, estuvo presente el reuso de componentes (Figura 3.23), técnica que ayudó a reducir costos de tiempo e implementación. La librería de código abierto IAXclient y clases propias de Qt facilitaron en gran medida la implementación del *softphone XoIP*.

Todo el sistema se implementó bajo un proyecto en QT Designer con el nombre de Tesis.pro, en el cual se implementaron las clases de diseño y las interfaces del sistema.

3.4.2. Implementación de las clases de diseño

Las Figuras 3.24-3.26 muestran las clases de diseño y sus respectivos componentes de implementación. En la Figura 3.25 se puede observar que para cada clase *boundary* no fue necesario un único componente de implementación; lo anterior se hizo con la finalidad de tener en un sólo archivo las clases relacionadas. Caso contrario fueron las clases *entity* y *control*, Figuras 3.24 y 3.26 respectivamente, en donde se creó un componente de implementación por cada clase de diseño con el objetivo de hacer una implementación que pueda evolucionar en cuanto a funcionalidad y estructura.

Todos los componentes de implementación fueron codificados en QT Designer mediante el lenguaje de programación C++. Una vez obtenidos y probados todos los componentes de implementación se procedió a generar el archivo ejecutable de la aplicación en el sistema operativo Linux en su distribución Ubuntu 6.06 (qmake -o Makefile Tesis.pro).

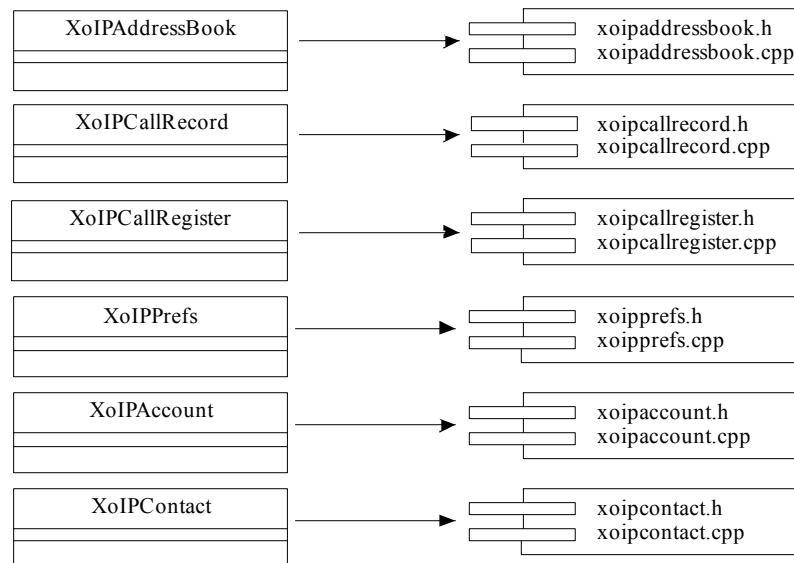


Figura 3.24. Evolución de las clases *Entity* de diseño en componentes de implementación.

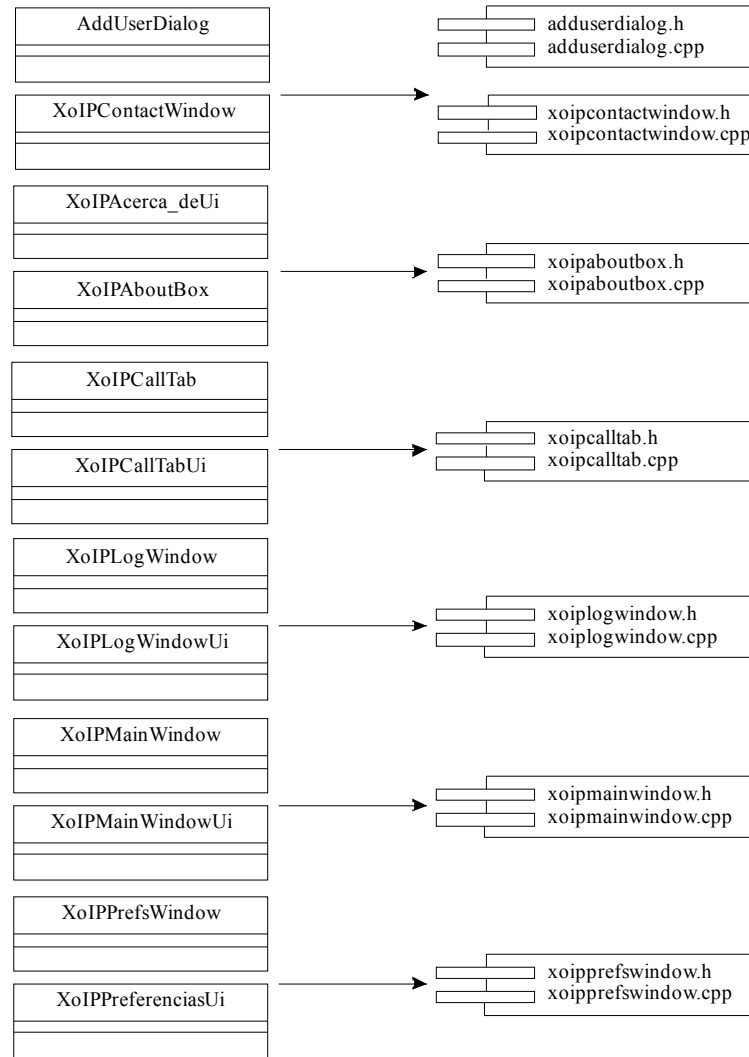


Figura 3.25. Evolución de las clases *Boundary* de diseño en componentes de implementación.

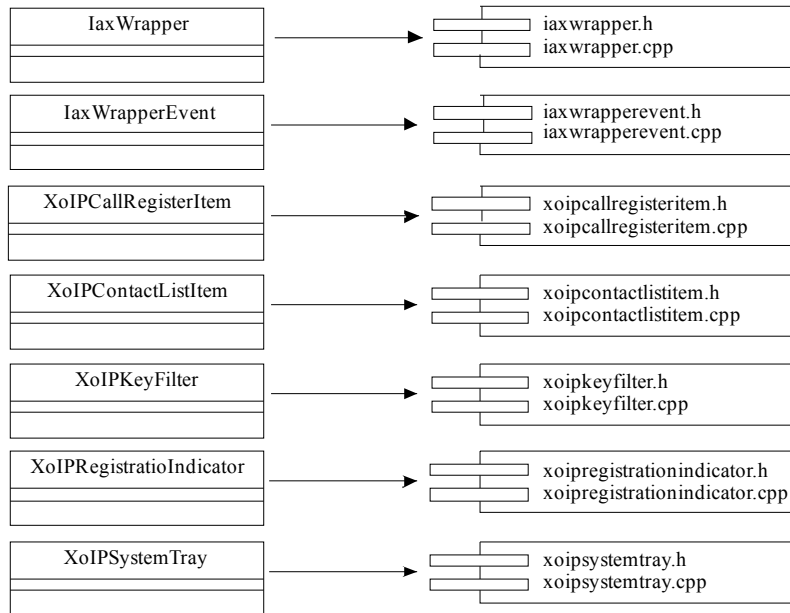


Figura 3.26. Evolución de las clases *Control* de diseño en componentes de implementación.

3.4.3. Asterisk y el *softphone XoIP*

Una vez que la aplicación funcionaba correctamente fue necesario configurar Asterisk mediante la creación de cuentas de usuario y la realización de pruebas de conexión desde la aplicación *softphone XoIP* con teléfonos dentro y fuera de las instalaciones de red de la Universidad Tecnológica de la Mixteca, para ello fue necesario conocer la sintaxis y las órdenes de Asterisk.

Debido a que Asterisk hace uso del mismo protocolo de comunicación VoIP que el *softphone XoIP*, no fue muy complicado entender la funcionalidad de éste, las dificultades se presentaron al momento de crear las cuentas de usuario, ya que éstas deben estar configuradas de manera que al ser utilizadas puedan establecer una conversación de voz aceptable.

3.5. Pruebas del *softphone XoIP*

La fase de pruebas consistió en un análisis y estudio detallado de las funcionalidades del *softphone XoIP* con la finalidad de demostrar que los objetivos planteados al inicio de este trabajo de tesis fueron logrados exitosamente.

Cada módulo del *softphone XoIP* fue sometido a pruebas con el fin de localizar errores de funcionalidad o de ejecución; para la realización de las pruebas se utilizaron dos tipos de técnicas:

- *Pruebas de caja blanca*: Consistieron en revisar detalladamente y depurar las líneas de código de cada módulo del sistema con la finalidad de encontrar errores de funcionamiento y de comportamiento para corregirlos posteriormente.
- *Pruebas de caja negra*: Consistieron en realizar pruebas de funcionamiento de cada módulo del sistema, enviando los parámetros de entrada necesarios y analizar los resultados.

Debido a que el sistema fue desarrollado dentro de los flujos de trabajo fundamentales del UP, cabe mencionar que dichas pruebas se realizaron repetidas veces hasta lograr los objetivos planteados. Todos los errores encontrados fueron corregidos inmediatamente hasta lograr la calidad de ejecución y funcionamiento esperados.

Para lograr el objetivo de este trabajo de tesis se puso en funcionamiento una PBX en software, a la que se ha venido haciendo referencia como Servidor/IP PBX. Para ello se aprendió el lenguaje de configuración de la PBX, el cual es de código abierto y maneja un lenguaje de *script* para realizar las configuraciones necesarias y poner en funcionamiento las soluciones a las necesidades establecidas.

Las configuraciones realizadas, para poner en funcionamiento y ejecución al Servidor, se basaron en siete archivos: *asterisk.conf*, *extensions.conf*, *meetme.conf*, *musiconhold.conf*, *sip.conf*, *voicemail.conf* y *zapata.conf*. El archivo *asterisk.conf* contiene la información para mantener al Servidor/ IP PBX funcionando, mientras que el archivo *extensions.conf* define la forma en que se deben manejar las llamadas, este archivo es fundamental para el funcionamiento del PBX (Figura 3.27).

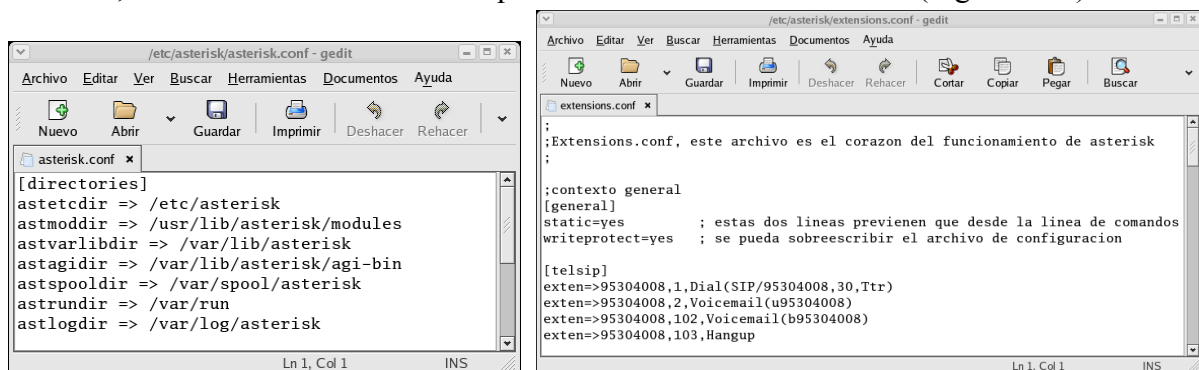


Figura 3.27. Configuración de los archivos *asterisk.conf* y *extensions.conf*.

La Figura 3.28 muestra el procedimiento de configuración del Servidor/IP PBX.

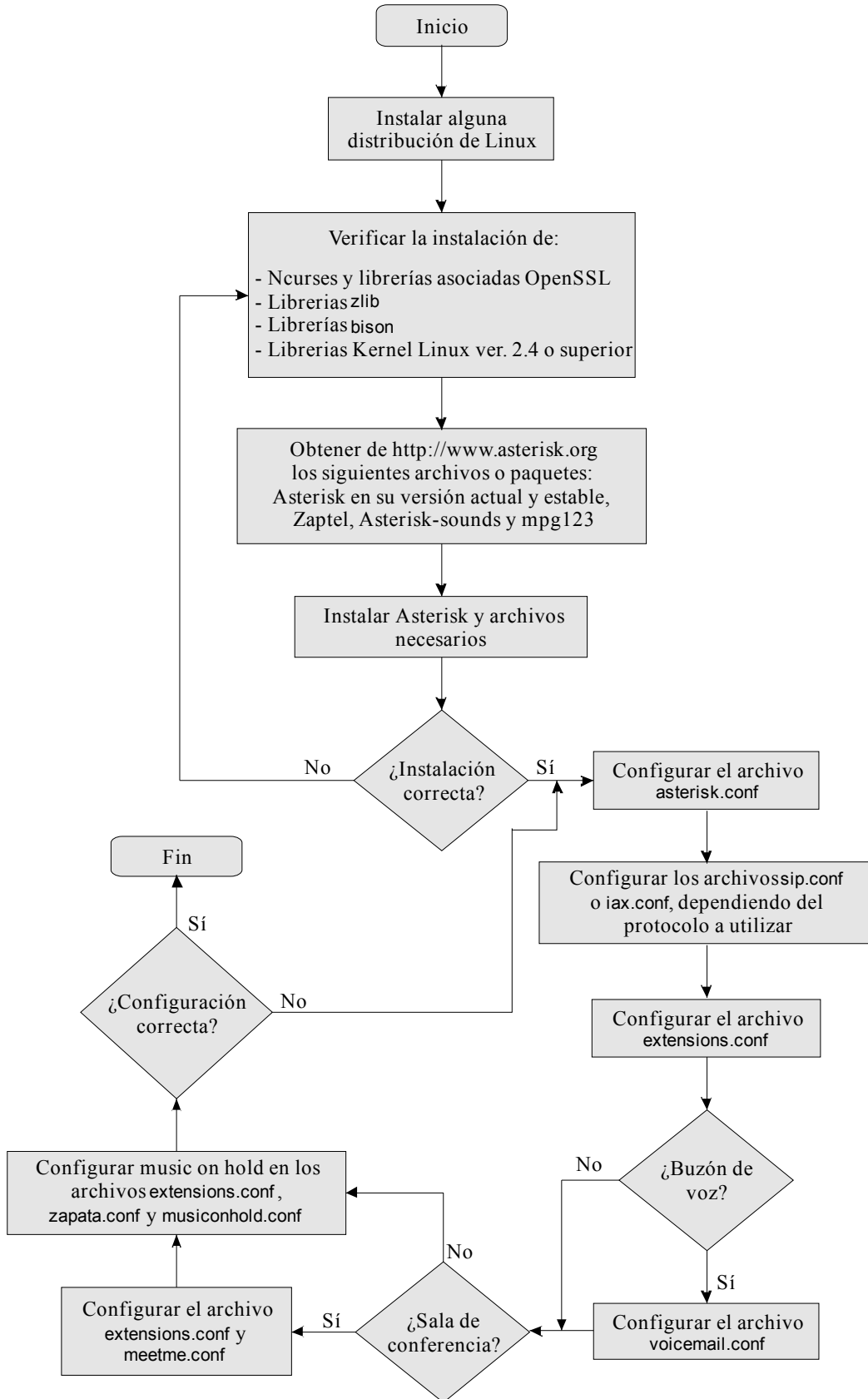
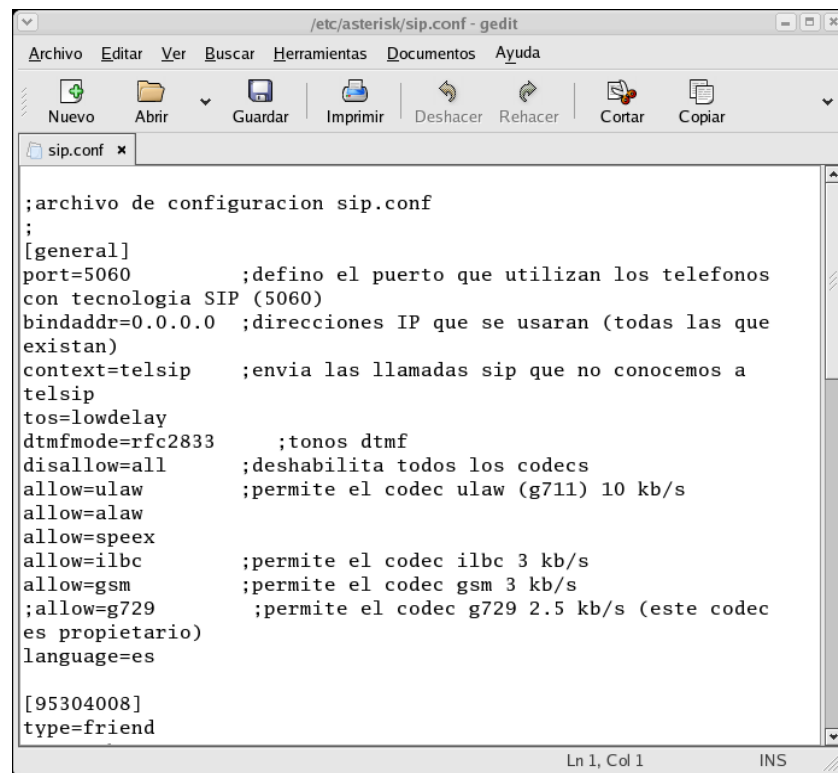


Figura 3.28. Diagrama de instalación y configuración del Servidor/IP PBX.



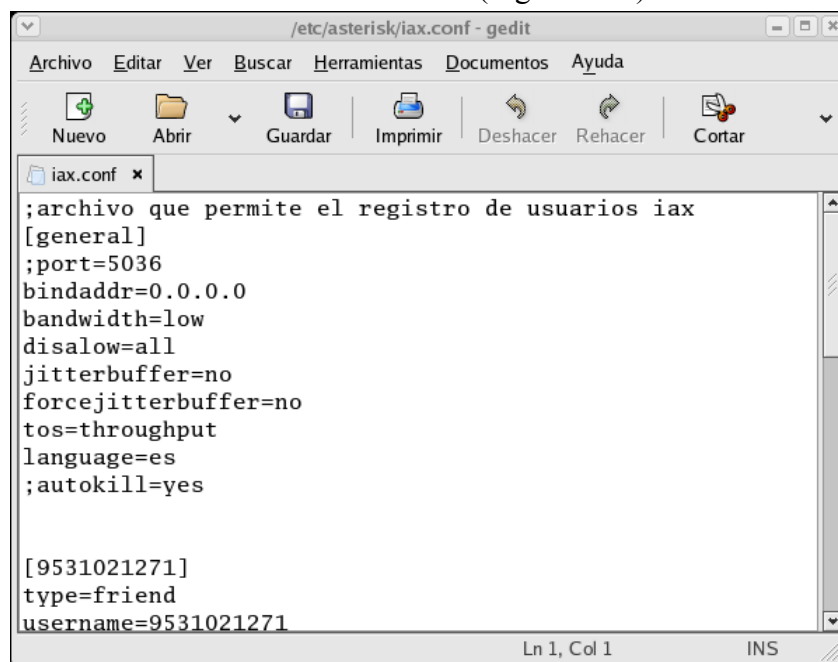
```
/etc/asterisk/sip.conf - gedit
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
Nuevo  Abrir  Guardar  Imprimir  Deshacer  Rehacer  Cortar  Copiar
sip.conf x
;archivo de configuracion sip.conf
;
[general]
port=5060           ;defino el puerto que utilizan los telefonos
                    ;con tecnologia SIP (5060)
bindaddr=0.0.0.0   ;direcciones IP que se usaran (todas las que
                    ;existan)
context=telsip     ;envia las llamadas sip que no conocemos a
                    ;telsip
tos=lowdelay
dtmfmode=rfc2833   ;tonos dtmf
disallow=all       ;deshabilita todos los codecs
allow=ulaw         ;permite el codec ulaw (g711) 10 kb/s
allow=alaw
allow=speex
allow=ilbc         ;permite el codec ilbc 3 kb/s
allow=gsm          ;permite el codec gsm 3 kb/s
;allow=g729        ;permite el codec g729 2.5 kb/s (este codec
                    ;es propietario)
language=es

[95304008]
type=friend
Ln 1, Col 1  INS
```

Figura 3.29. Archivo de configuración sip.conf.

La Figura 3.29 muestra la configuración del archivo sip.conf si se requiere hacer uso del protocolo SIP, mientras que la Figura 3.30 muestra la configuración del archivo iax.conf para utilizar el protocolo IAX, objetivo principal del presente trabajo.

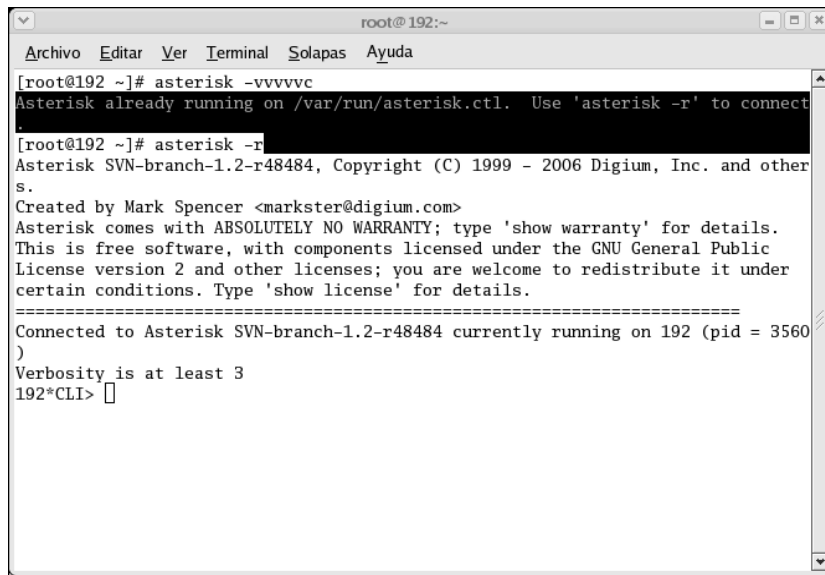
Se realizó el mismo procedimiento de configuración a los demás archivos involucrados. Una vez realizadas las configuraciones correspondientes, el paso siguiente fue comprobar que el Servidor/IP PBX se encontrara funcionando correctamente (Figura 3.31).



```
/etc/asterisk/iax.conf - gedit
Archivo  Editar  Ver  Buscar  Herramientas  Documentos  Ayuda
Nuevo  Abrir  Guardar  Imprimir  Deshacer  Rehacer  Cortar
iax.conf x
;archivo que permite el registro de usuarios iax
[general]
;port=5036
bindaddr=0.0.0.0
bandwidth=low
disallow=all
jitterbuffer=no
forcejitterbuffer=no
tos=throughput
language=es
;autokill=yes

[9531021271]
type=friend
username=9531021271
Ln 1, Col 1  INS
```

Figura 3.30. Archivo de configuración iax.conf.

A terminal window titled 'root@192:~' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Terminal', 'Solapas', and 'Ayuda'. The terminal shows the execution of 'asterisk -vvvvv', which results in a message that Asterisk is already running. Subsequently, 'asterisk -r' is executed, displaying version information and a license notice. The terminal then shows the Asterisk CLI prompt '192*CLI>' with a cursor.

```
root@192:~  
[root@192 ~]# asterisk -vvvvv  
Asterisk already running on /var/run/asterisk.ctl. Use 'asterisk -r' to connect  
[root@192 ~]# asterisk -r  
Asterisk SVN-branch-1.2-r48484, Copyright (C) 1999 - 2006 Digium, Inc. and other  
s.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'show warranty' for details.  
This is free software, with components licensed under the GNU General Public  
License version 2 and other licenses; you are welcome to redistribute it under  
certain conditions. Type 'show license' for details.  
=====  
Connected to Asterisk SVN-branch-1.2-r48484 currently running on 192 (pid = 3560  
)  
Verbosity is at least 3  
192*CLI> █
```

Figura 3.31. Servidor/IP PBX en ejecución.

4. Resultados

Una vez configurado y probado el correcto funcionamiento del Servidor/IP PBX, se ejecutó la aplicación desarrollada mediante el procedimiento que muestra la Figura 4.1.

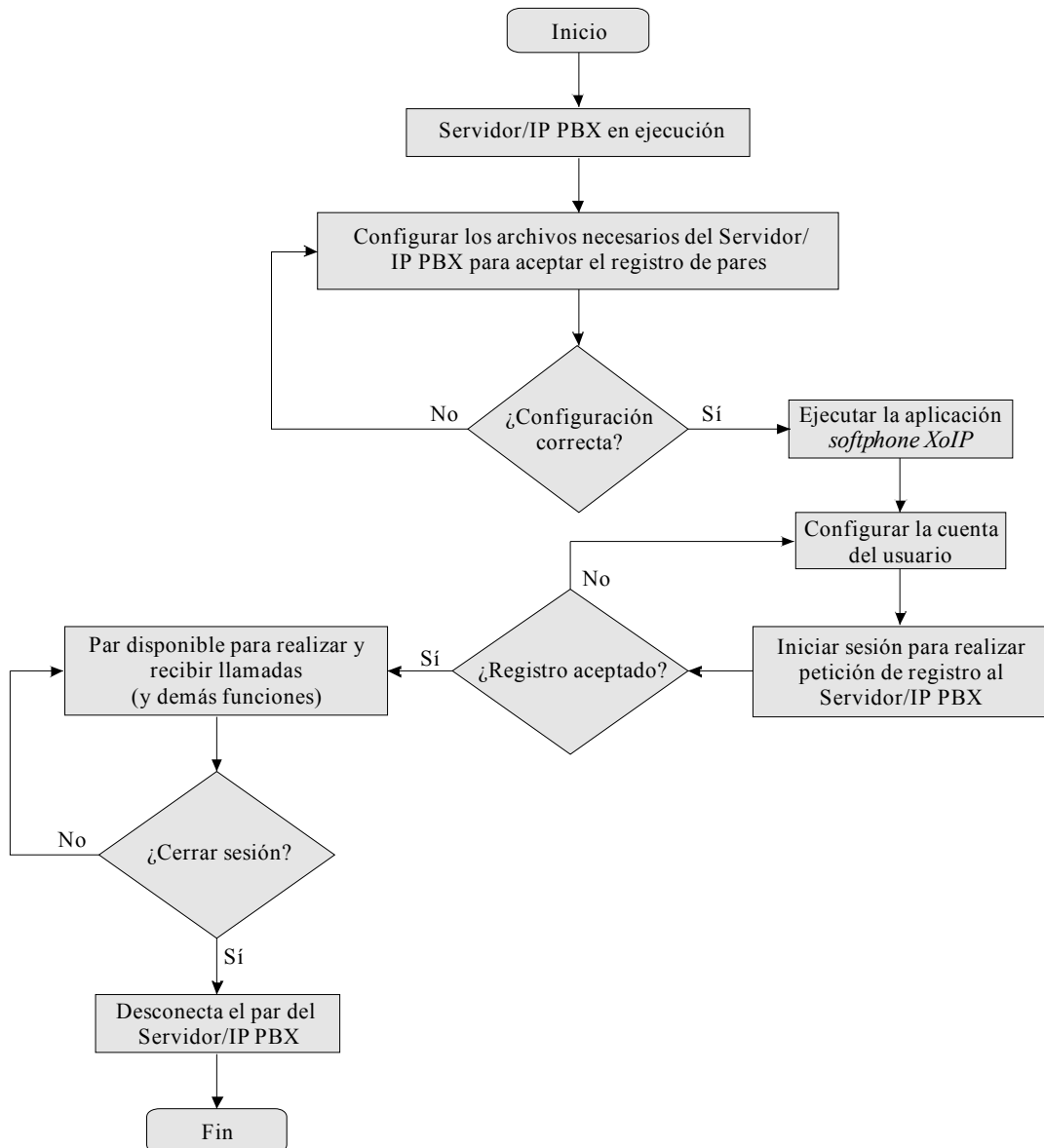


Figura 4.1. Diagrama de flujo general sobre el funcionamiento de XoIP.

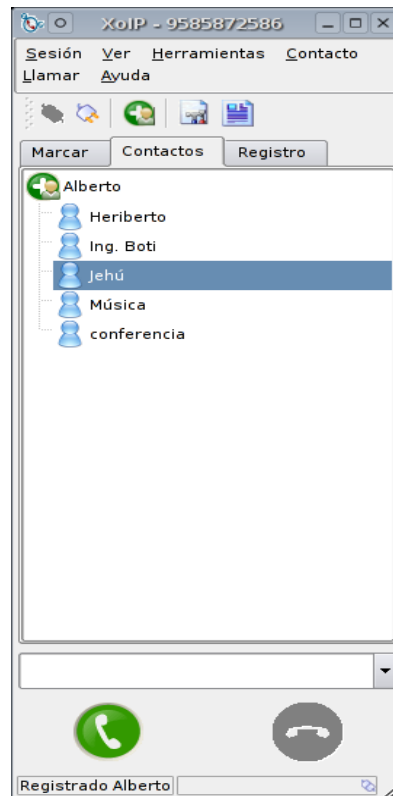


Figura 4.2. Cliente XoIP registrado en el Servidor/IP PBX.

Realizado el procedimiento anterior, la aplicación del *softphone* XoIP inició sesión para registrarse en el Servidor/IP PBX (Figura 4.2) y para comprobar su funcionamiento se estableció una llamada telefónica con otro cliente IAX registrado en el mismo Servidor (Figura 4.3).

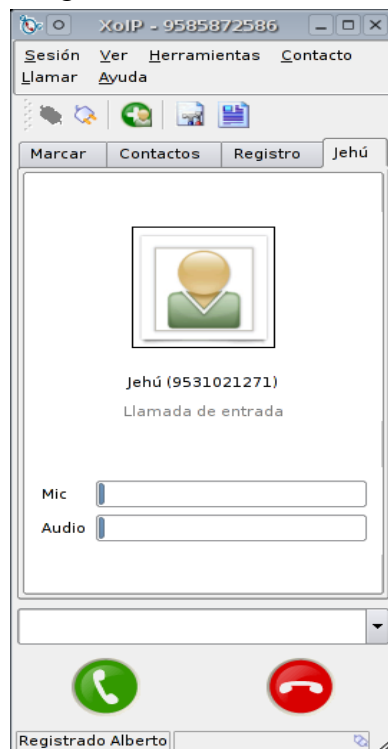


Figura 4.3. Establecimiento de llamada con otro cliente IAX registrado en el Servidor/IP PBX.



Figura 4.4. Aplicación que usa el protocolo SIP y que se encuentra registrada en el Servidor/IP PBX.

Cabe mencionar que el trabajo realizado no sólo demuestra el funcionamiento de aplicaciones que hacen uso del protocolo IAX, si no que también demuestra el funcionamiento de las que utilizan el protocolo SIP. En la Figura 4.4 se muestra la capacidad del Servidor/IP PBX de aceptar peticiones de registros de clientes SIP y de realizar llamadas entre clientes SIP y clientes IAX (Figura 4.5).



Figura 4.5. Establecimiento de llamada de un cliente SIP con un cliente IAX.

Otras de las pruebas que se realizaron, es que el *softphone XoIP* no sólo funciona con *softphones* sino que tiene la capacidad de adoptar el registro de clientes con teléfonos convencionales o clientes que hacen uso de teléfonos IP en hardware como muestran las Figuras 4.6-7.



Figura 4.6. Teléfono IP registrado en el Servidor/IP PBX.

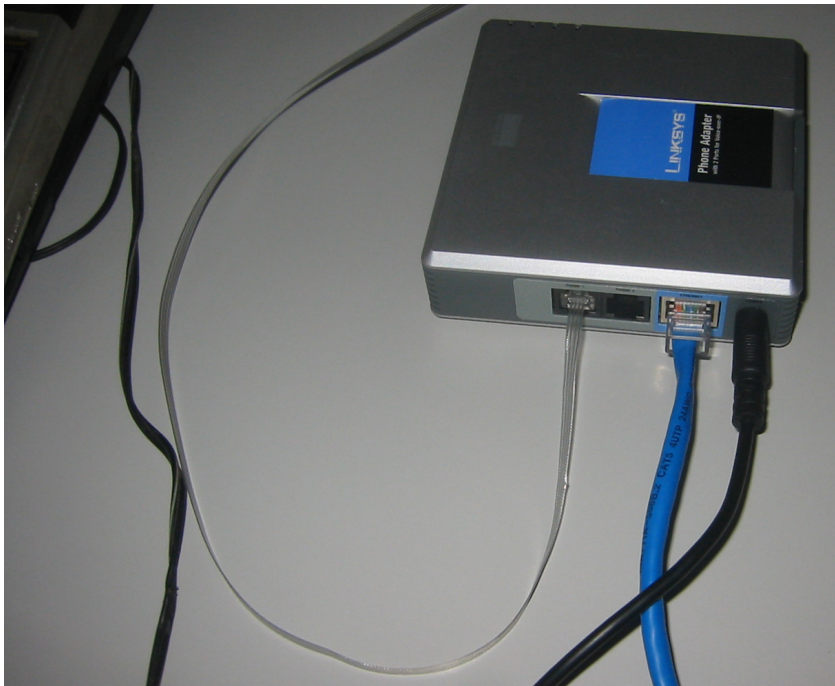


Figura 4.7. Adaptador telefónico utilizado como puente para el registro con el Servidor IP/PX.

El Anexo A presenta el manual de usuario del *softphone XoIP*.

5. Conclusiones y líneas futuras de investigación

El presente trabajo de tesis fue desarrollado bajo varios campos de investigación: tecnología orientada a objetos, telecomunicaciones de voz sobre IP y reuso de componentes, los cuales en la actualidad son campos de investigación de gran relevancia para los desarrolladores de software.

Con la finalidad de cumplir con los objetivos planteados, se llevaron a cabo las siguientes actividades:

- Estudio formal de los elementos involucrados en los sistemas de telecomunicaciones sobre IP (Capítulo 1).
- Investigación y estudio de los diferentes codecs existentes (Subcapítulo 1.5).
- Investigación y estudio de los principales protocolos de establecimiento de llamada empleados en tecnología VoIP (Capítulo 2).
- Configuración y puesta en funcionamiento de una centralita VoIP (Servidor/IP PBX) para dar soporte a *softphones* con tecnología SIP o IAX (Subcapítulo 3.5).
- Probar el funcionamiento de diversos *softphones* existentes con el Servidor/IP PBX (Apartado 3.2.1).
- Modelar el diseño del *softphone XoIP* mediante el UML de acuerdo a los requerimientos establecidos (Capítulo 3).
- Diseñar la GUI del *softphone XoIP* (Subcapítulo 3.3).
- Documentar el desarrollo del *softphone XoIP* mediante la utilización de herramientas de modelado y realizar un documento que acompañe a la aplicación final, para que el usuario obtenga los conocimientos técnicos y teóricos sobre el funcionamiento de la herramienta desarrollada (Capítulos 3 y 4, Anexo A y documentos incluidos en el CD adjunto).

Cabe mencionar como principales aportaciones del trabajo:

- Modelado del *softphone XoIP* en UML con base a los flujos de trabajo del UP.
- Estudio formal del protocolo IAX (Subcapítulo 2.4).
- Estudio y configuración del Servidor/IP PBX con soporte para buzón de voz, salas de conferencia por voz y el uso de DTMF.
- Ejemplos de los archivos de configuración principales del Servidor/IP PBX.
- Capacidad del establecimiento de llamadas entre teléfonos basados en los protocolos de establecimiento de llamada SIP e IAX.

Con el Servidor/IP PBX en software se demuestra la capacidad potencial de invertir en PBXs de bajo costo y con una plataforma operativa de código abierto, lo cual disminuye los costos que

representan las licencias software y genera una creciente capacidad tecnológica, puesto que se hace uso de una infraestructura existente capaz de adoptar nuevas tecnologías.

De lo anterior se puede concluir lo siguiente:

- Con el uso de VoIP, las empresas e instituciones pueden reducir los costos de comunicación telefónica.
- Con el uso del *softphone XoIP* se puede acceder desde cualquier equipo de cómputo que tenga una conexión a Internet al conmutador telefónico de la empresa o institución para la cual se labora.
- Se puede administrar el ancho de banda, con la finalidad de destinar cierto porcentaje a los paquetes de datos y cierto porcentaje a los paquetes de voz, inclusive establecer prioridad entre ellos y dar cierta QoS a la red de datos en general.
- De acuerdo a la investigación realizada se pueden conectar teléfonos convencionales a través de un ATA o FXS al Servidor/IP PBX en funcionamiento.
- Se tienen los conocimientos necesarios para retomar el tema y profundizarlo más con el objetivo de conectar el Servidor/IP PBX en software a una centralita en hardware comercial con tarjetas FXO o alguna otra tarjeta que permita establecer un puente de comunicación con la PSTN.
- Con la información de este trabajo de tesis se tienen las bases para conectar más de dos centralitas IP permitiendo la comunicación entre campus y universidades sin costo alguno.
- Se tiene la información necesaria que permitirá crear extensiones telefónicas en software y en hardware, todas estas comunicadas a través de Internet con los teléfonos IP H.323 con los que cuenta la Universidad Tecnológica de la Mixteca.

Con la información que proporciona este trabajo de tesis, se pueden realizar trabajos adicionales que vendría a favorecer a la red de Universidades del SUNEО, el objetivo es comunicar telefónicamente por IP a todos los campus y universidades generando un ahorro económico sustancial.

Otras de las tareas por realizar a futuro es utilizar la misma red de datos para el envío de Fax por IP (FoIP), con la finalidad de explorar al máximo la infraestructura de red del ISP.

Finalmente, el *softphone XoIP* es una herramienta que ayuda a conocer la tecnología VoIP y en beneficio de los usuarios finales puede considerarse como una solución de bajo costo para brindar comunicaciones de voz sobre redes IP. Asimismo, los desarrolladores de software pueden conocer el funcionamiento e identificar los componentes involucrados en los sistemas de telecomunicaciones por IP.

6. Bibliografía

- [1] Anaya, R.: “Un acercamiento a la reutilización en ingeniería de software”, Revista Universidad, Eafit No. 114, pp. 49-63, 1999.
- [2] Biggerstaff, T.: “A Perspective of Generative Reuse”, Annals of Software Engineering, Vol. 5, pp. 169-226, 1998.
- [3] Black, U.: Voice over IP, Prentice Hall, Second edition, 2002.
- [4] Booch, G., Rumbaugh, J. y Jacobson, I.: El Lenguaje Unificado de Modelado, Addison Wesley, 1999.
- [5] García, M.: “VoIP: una puerta hacia la convergencia”, Documento técnico del Departamento de Lenguajes y Sistemas Informáticos, Universidad de Vigo, España, pp. 1-14, Junio 1999.
- [6] Gavilán, J.: “Voz IP: Presente y Futuro de las comunicaciones de Voz”, Ponencias Internet 99, pp. 1-5, 1999.
- [7] Gomillion, D. & Dempster, B.: Building Telephony Systems with Asterisk, PACKT Publishing Ltd., 2005.
- [8] Gonzáles, A.: “Integración de un modelo de proceso sistemático en el desarrollo de software educativo”, Tesis de Doctorado, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Vigo, España, pp. 147-153, .
- [9] González, A.: Estudio del protocolo IEEE 488 mediante el desarrollo de una herramienta de simulación, Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, Abril 2003.
- [10] IBM: “Rational Unified Process: Best practices for software development teams”, IBM Staff, pp. 1-17, July 2005.
- [11] Jacobson, I., Booch, G. y Rumbaugh, J.: El Proceso Unificado de Desarrollo de Software, Addison Wesley, 2000.
- [12] Krutchen, P.: The Rational Unified Process; An Introduction, Addison Wesley Longman, 2000.
- [13] Matías, I.: Herramienta Case para la generación de código C++ a partir de diagramas de clase UML, Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, Julio 2003.
- [14] Meyer, B.: Construcción de Software Orientado a Objetos, Prentice Hall, Segunda edición, 1999.
- [15] Pérez, F.: Desarrollo de componentes de software local y distribuido bajo la plataforma COM-SOAP, que encapsule las operaciones matriciales de MATLAB, Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, Diciembre 2005.
- [16] Presman, R.: Ingeniería del Software; Un enfoque práctico, Mc-Graw-Hill, Cuarta edición,

1998.

- [17] Rumbaugh, J., Jacobson, I. y Booch, G.: El lenguaje Unificado de Modelado. Manual de Referencia, Addison Wesley, 2000.
- [18] Sametinger J.: Software Engineering with Reutilizable Components, Springer Verlag, Software Components, 1997.
- [19] Spencer, M.: “IAX: Inter-Asterisk eXchange Versión 2”, Internet-Draft, pp. 1-187, September 2006.
- [20] Stallings, W.: Comunicaciones y Redes de Computadores, Pearson, Sexta edición, 2000.
- [21] Tanenbaum, A.: Computer Networks, Prentice Hall, Fourth edition, 2003.
- [22] Van Meggelen, J., Smith, J. & Madsen, L.: Asterisk™: The Future of Telephony, O’Reilly, 2005.

URLs

- [URL1] <http://compare.ozvoip.com/codecs.php> “Página Web destinada a proporcionar información sobre distintos codecs”, Enero 2007.
- [URL2] <http://doc.trolltech.com/> “Página Web para desarrolladores de software sobre el Ambiente QT-Designer”, 2006.
- [URL3] <http://iaxclient.sourceforge.net/iaxcomm/> “Página Web del *softphone* Iaxcomm”, 2006.
- [URL4] <http://sourceforge.net/projects/iaxclient/> “Página Web del proyecto *open source* de la librería IAXClient”, 2006.
- [URL5] <http://vozdigital.org/> “Foro destinado a VoIP y Asterisk”, 2006.
- [URL6] <http://www.asterisk.org/> “Página Web dedicada a Asterisk”, 2006.
- [URL7] <http://www.asteriskguru.com/idefisk/> “Página oficial del *softphone* IdeFisk”, 2006.
- [URL8] <http://www.gtas.dicom.unican.es/> “Página Web de la Universidad de Cantabria”, 2005.
- [URL9] <http://www.ilbcfreeware.org/> “Página Web dedicada al proyecto iLBC *freeware* sobre el codec iLBC”, 2006.
- [URL10] <http://www.laser.com/dante/diax/diaxhlp.htm> “Página Web del *softphone* Diax”, 2006.
- [URL11] <http://www.openh323.org> “Página del proyecto de código abierto H.323”, 2005.
- [URL12] <http://www.sipfoundry.org> “Página Web del código abierto VoIP”, 2005.
- [URL13] <http://www.skype.com> “Página Web del *softphone* Skype dedicada a la venta de minutos para la realización de llamadas a teléfonos fijos y celulares a través de VoIP”, 2005.
- [URL14] <http://www.speex.org/> “Página Web de la comunidad del codec Speex”, 2006.
- [URL15] <http://www.usergioarboleda.edu.co/grupointernet/> “Página Web del Grupo Internet, Universidad Sergio Arboleda, Octubre 2005.
- [URL16] <http://www.voipforo.com/codec/codecs.php> “Página Web que presenta una comparativa de codecs y toda información relacionada a la Voz sobre IP”, Enero 2007.
- [URL17] <http://www.voip-info.org/wiki-Codecs> “Página Web destinada a la distribución de información sobre codecs VoIP”, 2006.
- [URL18] <http://www.voipreview.org> “Conceptos y *Carriers* VoIP”, Noviembre 2005.
- [URL19] <http://www.whichvoip.com/> “Página Web de conceptos y proveedores VoIP”, 2005.
- [URL20] <http://www.xten.com> “Página Web de la firma CounterPath cuyo objetivo es el desarrollo de *softphones* profesionales”, 2005.

Lista de acrónimos

- ABS (*Analysis by Synthesis*, Análisis por síntesis)
- ACM (*Address Complete Message*, Mensaje de dirección completa)
- ADPCM (*Adaptative Differential Pulse Code Modulation*, Modulación por codificación de impulsos diferencial adaptativa)
- AES (*Advanced Encryption Standard*, Estándar de cifrado avanzado)
- ATA (*Analog Telephone Adapter*, Adaptador de teléfono analógico)
- ATM (*Asynchronous Transfer Mode*, Modo de transferencia asíncrona)
- AVP (*Audio-Video Profile*, Perfil de audio-video)
- B-ISDN (*Broadband-Integrated Services Digital Network*, Red digital de servicios integrados de banda ancha)
- CASE (*Computer Aided Software Engineering*, Ingeniería de software asistida por computadora)
- CELP (*Code Excited Linear Prediction*, Predicción lineal excitada por código)
- DiffServ (*Differentiated Services*, Servicios diferenciados)
- DNS (*Domain Name System*, Sistema de nombres de dominio)
- DoS (*Denial of Service*, Negación de servicio)
- DPCM (*Differential Pulse Code Modulation*, Modulación diferencial por código de impulsos)
- DSP (*Digital Signal Processor*, Procesador digital de señales)
- DTMF (*Dual Tone Multifrequency*, Multifrecuencia de doble tono)
- DTX (*Discontinuous Transmission*, Transmisión discontinua)
- ESP (*Enhanced Services Provider*, Proveedor de servicios mejorados)
- FCC (*Federal Communications Commission*, Comisión Federal de Comunicaciones)
- FoIP (*Fax over Internet Protocol*, Fax sobre el protocolo de Internet)
- FR (*Frame Relay*, Retransmisión de tramas)
- FXO (*Foreign eXchange Office*, Oficina de intercambio foráneo)
- FXS (*Foreign eXchange Subscriber*, Abonado de intercambio foráneo)
- GSM (*Global System for Mobile Communications*, Sistema global de comunicaciones para móviles)
- GUI (*Graphics User Interface*, Interfaz gráfica de usuario)
- HTTP (*Hypertext Transfer Protocol*, Protocolo de transferencia de hipertexto)
- IAM (*Initial Address Message*, Mensaje de dirección inicial)

IAX (*Inter Asterisk eXchange Protocol*, Protocolo de intercambio entre Asterisk)
IEs (*Information Elements*, Elementos de información)
IETF (*Internet Engineering Task Force*, Grupo de trabajo en ingeniería de Internet)
ILBC (*Internet Low Bitrate Codec*, Codec para Internet de baja velocidad)
IP (*Internet Protocol*, Protocolo de Internet)
IPDC (*Internet Protocol Device Control*, Protocolo de Internet para el control de dispositivos)
IPv4 (*Internet Protocol version 4*, Versión 4 del protocolo de Internet)
IPv6 (*Internet Protocol version 6*, Versión 6 del protocolo de Internet)
ISDN (*Integrated Services Digital Network*, Red digital de servicios integrados o RDSI)
ISP (*Internet Service Provider*, Proveedor de servicios de Internet)
ISUP (*ISDN User Part*, Parte del usuario ISDN)
ITU (*International Telecommunication Union*, Unión Internacional de Telecomunicaciones)
IVR (*Interactive Voice Response*, Respuesta interactiva de voz)
LAN (*Local Area Network*, Red de área local)
LEC (*Local Exchange Carrier*, Portadora de intercambio local)
LPC (*Linear Prediction Coding*, Código de predicción lineal)
MCU (*Multipoint Control Unit*, Unidad de control multipunto)
Megaco (*Media Gateway Controller*, Controlador de pasarelas de medias)
MGCP (*Media Gateway Control Protocol*, Protocolo de control de pasarelas de medias)
MIME (*Multipurpose Internet Mail Extension*, Extensiones de correo Internet multipropósito)
NAT (*Network Address Translation*, Traducción de direcciones de red)
N-ISDN (*Narrowband-Integrated Services Digital Network*, Red digital de servicios integrados de banda estrecha)
NTP (*Network Time Protocol*, Protocolo de sincronización de tiempos de red)
OOP (*Object Oriented Programming*, Programación orientada a objetos)
OPC (*Oriented Programming Components*, Programación orientada a componentes)
OSI (*Open System Interconnection*, Interconexión de sistemas abiertos)
PBX (*Private Branch Exchange*, Sistema telefónico privado)
PCM (*Pulse Code Modulation*, Modulación por impulsos codificados)
PSTN (*Public Switched Telephone Network*, Red pública de telefónica conmutada o red telefónica tradicional o básica)
PTT (*Push To Talk*, Presiona para hablar)
QoS (*Quality of Service*, Calidad de servicio)
RSVP (*Resource Reservation Setup Protocol*, Protocolo de reserva de recursos)
RTCP (*Real-Time Control Protocol*, Protocolo de control en tiempo real)
RTP (*Real-Time Transfer Protocol*, Protocolo de transferencia en tiempo real)
SDP (*Session Description Protocol*, Protocolo de descripción de la sesión)
SGMP (*Simple Gateway Control Protocol*, Protocolo de control de pasarelas simples)
SIP (*Session Initiation Protocol*, Protocolo de inicio de sesión)
SMTP (*Simple Mail Transfer Protocol*, Protocolo simple de transferencia de correo electrónico)

SS7 (*Signaling System 7*, Sistema de señalización por canal común No. 7)
STDM (*Statistical Time Division Multiplexing*, Multiplexación estadística por división de tiempo)
TCP (*Transmission Control Protocol*, Protocolo de control de transmisión)
TDM (*Time Division Multiplexing*, Multiplexación por división de tiempo)
Telco (*Telephone Company*, Compañía telefónica)
TLS (*Transport Layer Security*, Seguridad en la capa de transporte)
UDP (*User Datagram Protocol*, Protocolo de datagrama del usuario)
UML (*Unified Model Language*, Lenguaje unificado de modelado)
UNI (*User Network Interface*, Interfaces de usuario a red)
UP (*Unified Process*, Proceso unificado)
URI (*Uniform Resource Identifier*, Identificador de recurso uniforme)
URL (*Uniform Resource Locator*, Localizador de recurso uniforme)
VAD (*Voice Activity Detection*, Actividad de detección de voz)
VBR (*Variable Bitrate Operations*, Operaciones de velocidad de bit variable)
VoATM (*Voice over ATM*, Voz sobre ATM)
VoFR (*Voice over Frame Relay*, Voz sobre Frame Relay)
VoIP (*Voice over Internet Protocol*, Voz sobre el protocolo de Internet)
VPN (*Virtual Private Network*, Red privada virtual)
WAN (*Wide Area Network*, Red de área amplia)
XML (*Extensible Markup Language*, Lenguaje de marcas extensible)

A. Manual de usuario del *softphone XoIP*

El *softphone XoIP* es un teléfono en software desarrollado con el objetivo de realizar llamadas telefónicas a través de Internet. Fue implementado bajo la plataforma de Linux en la distribución *Ubuntu 6.06* y probado en otras como *Fedora Core 3* y *Debian Etch*.

El *softphone XoIP* soporta la transmisión de voz a través de Internet con base en la tecnología VoIP. A continuación se presenta el manual de usuario, el cual describe los siguientes puntos:

- Recursos necesarios para instalar el *softphone XoIP*.
- Compilar el *softphone XoIP*.
- Ejecutar *softphone XoIP*.
- Crear y eliminar cuentas de usuario.
- Descripción de la ventana principal.
- Agregar contactos.
- Realizar llamadas.
- *Log* de la aplicación.
- Ayuda.

A.1. Recursos necesarios para instalar el *softphone XoIP*

Para que el *softphone XoIP* pueda funcionar correctamente, es necesario cumplir con los siguientes requerimientos hardware:

- *Procesador*: Pentium II (como mínimo).
- *Memoria*: 128 MB (como mínimo).
- *Sistema operativo*: Linux.
- *Tarjeta de sonido*: SoundBlaster o equivalente de 16 bits.
- *Conexión a Internet*: alambrada (*wired*) o inalámbrica (*wireless*).

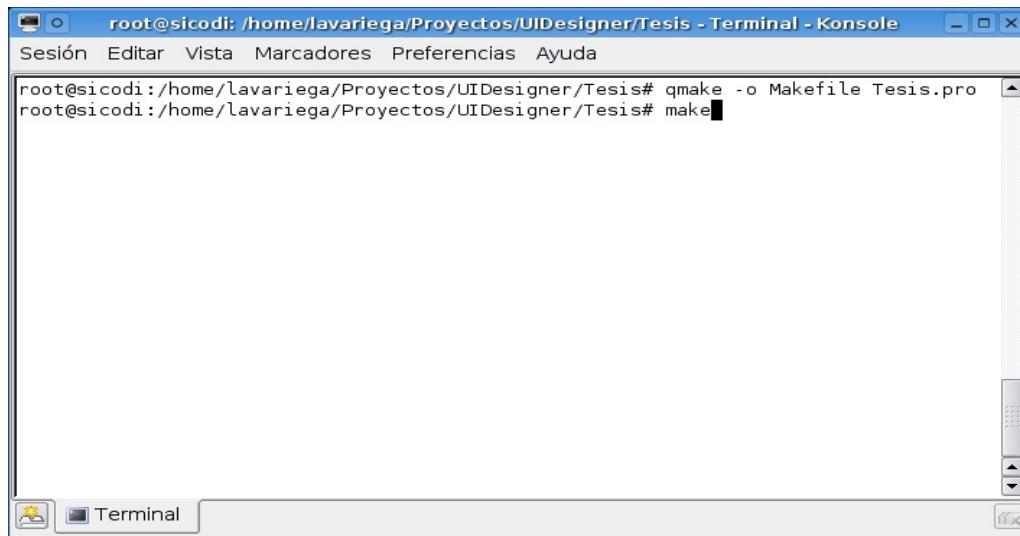


Figura A.1. Compilando el código fuente del *softphone XoIP*.

A.2. Compilación del *softphone XoIP*

Para aquellos usuarios o desarrolladores que cuenten con el código fuente del *softphone XoIP* y deseen compilarlo, a continuación se presenta la secuencia de pasos a seguir (Figura A.1):

- Abrir una terminal de la distribución Linux que se esté utilizando.
- Ejecutar `qmake -o Makefile Tesis.pro`¹ y presionar la tecla enter.
- Ejecutar la sentencia `make` y presionar la tecla enter para compilar el proyecto.

A.3. Ejecutar la aplicación *softphone XoIP*

Una vez que el proyecto se haya compilado satisfactoriamente, sin presencia de errores², se generará un archivo de nombre `xoip` en la ruta `../Tesis/bin`, donde la carpeta `Tesis` contiene el código fuente de la aplicación. En la carpeta `bin` se encuentra el archivo ejecutable de la aplicación; por lo tanto, para acceder a la ventana principal de la aplicación, es necesario seguir los siguientes pasos:

- Estando en la ruta `../Tesis`, cambiar a la carpeta `bin` mediante `cd bin`.
- Teclear `./xoip` seguida de la tecla enter.

Los pasos anteriores se deben realizar utilizando una terminal; si el usuario decide utilizar el entorno gráfico de la distribución, tiene que posicionarse en la ruta mencionada y dar doble clic sobre el archivo `xoip` o en su caso con el botón derecho del ratón elegir la opción abrir. La Figura A.2 muestra la ventana principal de *XoIP*.

A.4. Crear y eliminar cuentas de usuario

Una vez compilado y ejecutado el *softphone XoIP*, se pueden crear cuentas de usuario con la finalidad de establecer conexiones locales o remotas con el Servidor/IP PBX.

¹ El nombre de `Tesis.pro` es el nombre del proyecto, en el cual se incluyen todas las fuentes necesarias para la correcta compilación del *Softphone XoIP*.

² Cabe mencionar que el proyecto no tiene errores, por lo que si llega a mostrar alguno durante la fase de compilación puede deberse a otros factores como la ausencia de bibliotecas del sistema.



Figura A.2. Ventana principal del *softphone XoIP*.

Como muestra la Figura A.3, existe una cuenta por defecto, la cual contiene los parámetros necesarios para el establecimiento de conexión y/o registro con el Servidor/IP PBX. Para que se realice correctamente una conexión deben ser correctos los siguientes parámetros: Cuenta, Servidor IAX, ID de llamada, ID Número, Usuario y Contraseña.

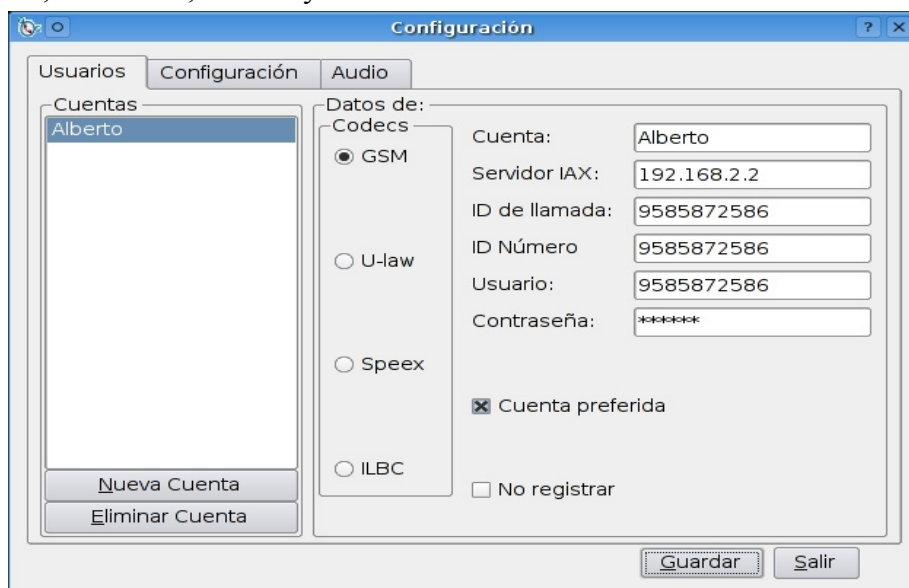


Figura A.3. Ventana para configurar las cuentas de usuario.

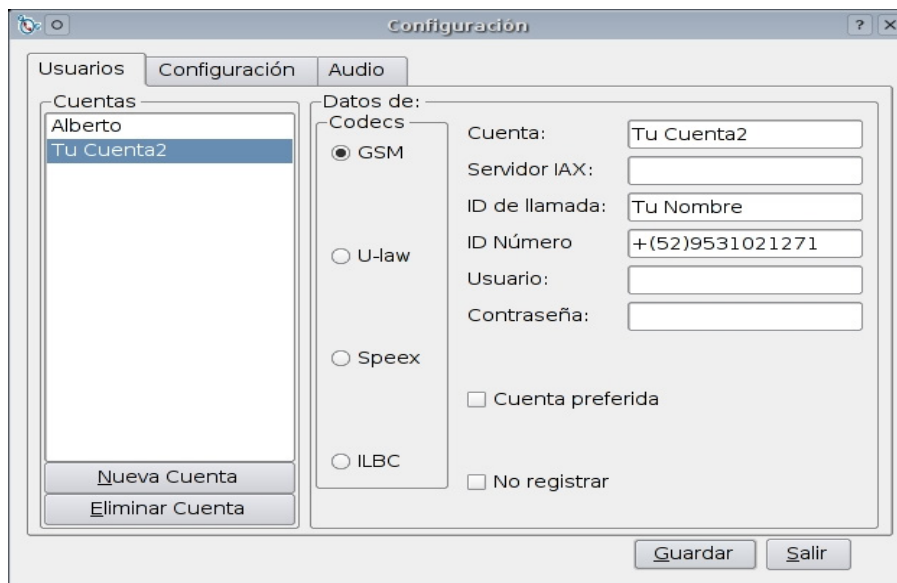



Figura A.4. Creación de una cuenta.

De acuerdo a la Figura A.3, se puede observar que la cuenta Alberto, además de los parámetros mencionados cuenta con la selección del codec GSM (pudiendo elegir entre GSM, U-law, Speex e ILBC). Existen dos pestañas más, Configuración y Audio, las cuales se describirán posteriormente.

Para crear una cuenta, se selecciona el menú Herramientas y la opción Configuración (o mediante el botón ). A continuación se presiona el botón Nueva Cuenta y la aplicación establecerá los datos por defecto (Figura A.4), los cuales se deben modificar de acuerdo con los proporcionados por el administrador del Servidor/IP PBX:

- *Cuenta*: en este campo se proporciona el nombre de la cuenta.
- *Servidor IAX*: se indica el nombre DNS o dirección IP del Servidor/IP PBX, por ejemplo: myserveriax.com o 10.0.0.1.
- *ID de llamada*: campo utilizado para proporcionar un identificador, por medio del cual los contactos puedan identificarse.
- *ID Número*: en este campo se proporciona el número asignado para acceder al Servidor/IP PBX.
- *Usuario*: es el nombre de usuario dado de alta en el Servidor/IP PBX.
- *Contraseña*: contraseña asignada al usuario.
- *Cuenta preferida*: indica la prioridad de la cuenta en caso de existir más cuentas.
- *No registrar*: indica sí la cuenta se registrará o no en el Servidor/IP PBX.
- *Codecs*: indica el codec por defecto que utilizará la cuenta.

A.5. Configuración de misceláneos

Una vez proporcionados los datos, se puede seleccionar cualquier opción de las que se muestran en la Figura A.5.

A.5.1. Configuración del audio

La Figura A.6 muestra las opciones básicas para el establecimiento de llamadas entre usuarios registrados en el Servidor/IP PBX, entre ellas se encuentra la selección de los dispositivos de audio de Entrada, Salida y Timbre, los cuales son necesarios para establecer una conversación de voz.

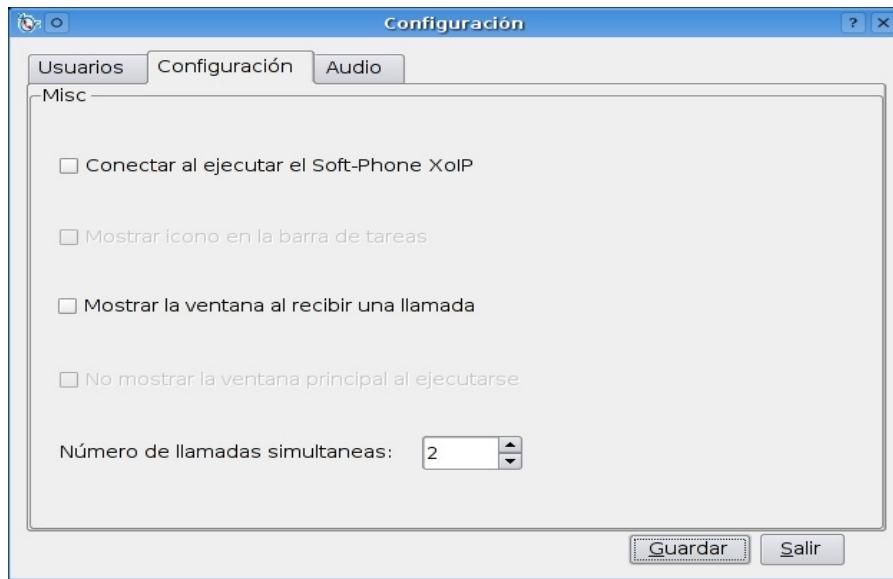


Figura A.5. Ventana de configuración de misceláneos.

Las opciones de calidad de servicio (QoS) hacen referencia a las características que proporcionan coherencia y claridad a la conversación de voz, mientras que las Opciones de Timbre habilitan o deshabilitan el timbre en llamadas Entrantes o Salientes (Figura A.6).

Una vez configuradas las opciones, se presiona el botón Guardar para almacenar los parámetros de la cuenta; si el usuario no proporciona los datos necesarios, la cuenta no puede ser almacenada y se muestra el mensaje de la Figura A.7.

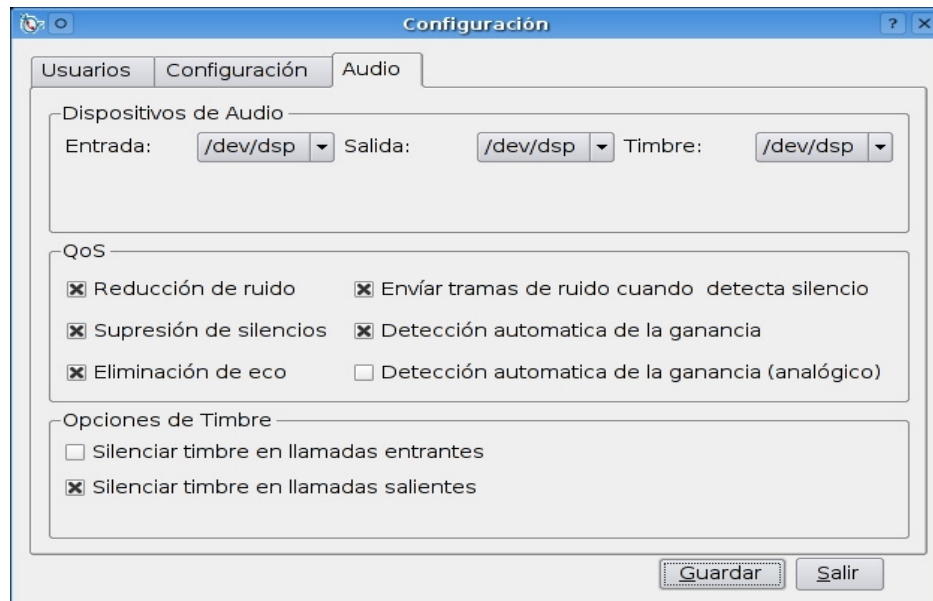


Figura A.6. Ventana de configuración para los dispositivos de audio.

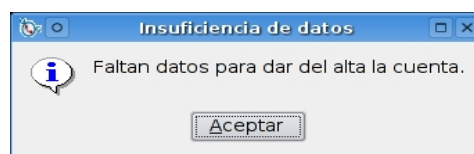


Figura A.7. Mensaje de insuficiencia de datos.

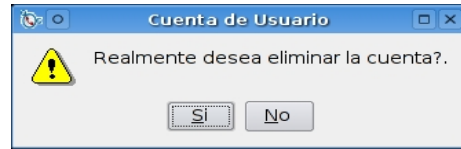





Figura A.8. Eliminando cuentas de usuario.

A.5.2. Eliminar cuenta de usuario

Para eliminar una cuenta de usuario en el *softphone XoIP*, se utiliza la opción Eliminar Cuenta (Figura A.3 y Figura A.4). Para confirmar la eliminación permanente de dicha cuenta, se presenta el mensaje de la Figura A.8.

A.6. Descripción de la ventana principal

Una vez que se han creado las cuentas de usuario, éste puede iniciar una sesión con su cuenta de preferencia. La Figura A.9 muestra la ventana principal de XoIP una vez que el usuario dio inicio a una sesión con su respectivo Servidor/IP PBX. Dicha ventana presenta seis opciones:

- Sesión:
 -  Iniciar: establece una conexión con el Servidor/IP PBX.
 -  Cerrar: elimina la conexión con el Servidor/IP PBX.
 -  Salir: elimina la conexión con el Servidor/IP PBX y cierra la aplicación del *softphone XoIP*.

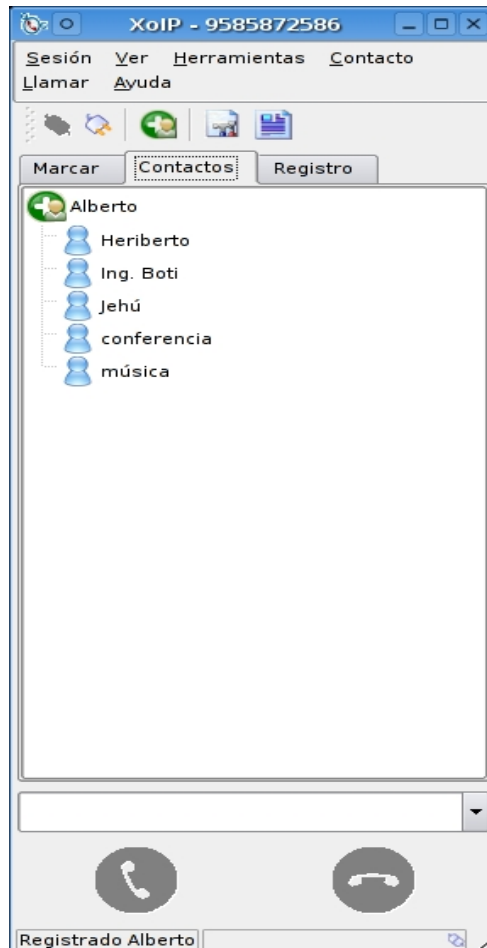












Figura A.9. Ventana principal, una vez que se ha iniciado sesión.

- Ver:
 - Panel de opciones: muestra las opciones más importantes.
 - Panel de marcación: para marcar a números no incluidos en la lista de Contactos.
 - Combo de marcado: visualiza el número al que se está marcando a través del panel de marcación.
 - Registro de llamadas: muestra un control de llamadas Entrantes, Salientes y Pérdidas.
- Herramientas:
 -  Limpiar registro de llamadas: elimina todo el registro de las llamadas Entrantes, Salientes y Pérdidas.
 -  Configuración: permite crear y eliminar cuentas de usuario, asignar el codec de preferencia, configurar los dispositivos de audio, etc.
 -  Ventana de registro: muestra los mensajes del *softphone XoIP* en tiempo de ejecución.
- Contacto:
 -  Agregar: agrega usuarios a la lista de Contactos.
 -  Eliminar: elimina usuarios de la lista de Contactos.
- Llamar:
 -  Marcar: solicita una petición de establecimiento de llamada al contacto seleccionado o al número que ha sido marcado a través del panel de marcación. También se utiliza para aceptar una petición de llamada por otro usuario.
 -  Contestar: aceptar el establecimiento de llamada solicitado por otro usuario registrado en el Servidor/IP PBX.
 -  Retener: se utiliza para detener el envío de paquetes de voz de la llamada establecida, pero no destruye la llamada.
 -  Restaurar: activa el envío de paquetes de voz de la llamada detenida.
 -  Colgar: termina la llamada actual.
- Ayuda:
 - Manual de usuario: presenta un manual de usuario del *softphone XoIP*.
 - Acerca de: muestra información del desarrollo del *softphone XoIP*.

La ventana principal del *softphone XoIP* muestra tres pestañas:

- Marcar: presenta un panel de marcación, a través del cual el usuario puede marcar a los contactos que no tiene datos de alta en su lista de Contactos (Figura A.12a).
- Contactos: muestra una lista de Contactos que el usuario tiene datos de alta, con la finalidad de facilitar el marcado (Figura A.12b).
- Registro: muestra una lista de todas las llamadas Entrantes, Salientes y Perdidas (Figura A.13b).

A.7. Agregar contactos

Para agregar un número de contacto, se utiliza la opción Agregar Contacto (Figura A.10), en la que se configura el alias y el número del contacto correspondiente.

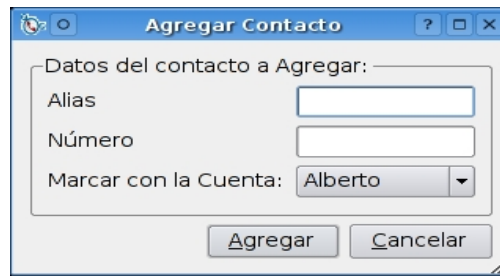


Figura A.10. Ventana Agregar Contacto.

Para dar de alta a dicho usuario es necesario presionar el botón Agregar, y si falta algún dato se muestra un mensaje de advertencia al usuario.

A.7.1. Actualizar datos de un contacto

Para actualizar los datos de un contacto, el *softphone* XoIP proporciona la opción Actualización de datos (Figura A.11).

A.8. Realizar llamadas

Para realizar una llamada a un usuario registrado es necesario haber iniciado una sesión. El marcado se puede realizar de dos maneras:

- Panel de marcación: por ejemplo si se desea marcar al número 95304009, se teclea cada dígito como muestra la Figura A.12a.
- Lista de Contactos: si el número a marcar está dado de alta en la lista de Contactos (Figura A.12b), sólo se debe seleccionar el contacto y presionar el botón Marcar o mediante un doble clic sobre el contacto.

A.8.1. Petición de llamada

En la Figura A.13a se muestra una pestaña adicional a la ventana principal, la cual indica que se está llamando al contacto correspondiente. Dicha pestaña muestra la información del usuario, así como el tiempo de duración de la llamada.

A.8.2. Registro de la llamada

Al término de la llamada se almacena un registro con el alias, fecha, hora y el tiempo de la llamada realizada con la finalidad de llevar un control de llamadas (Figura A.13b).

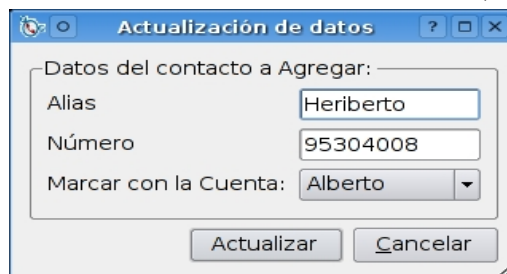


Figura A.11. Ventana para actualizar y corregir datos del contacto.

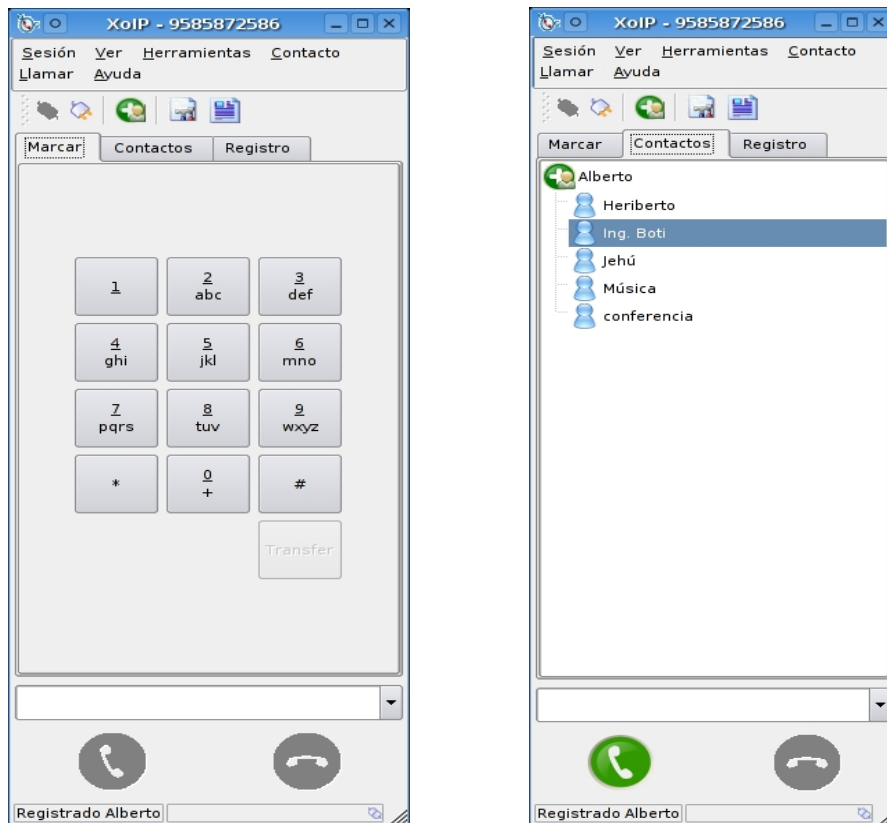


Figura A.12. a) Ventana del panel de marcación, b) Ventana de lista de contactos

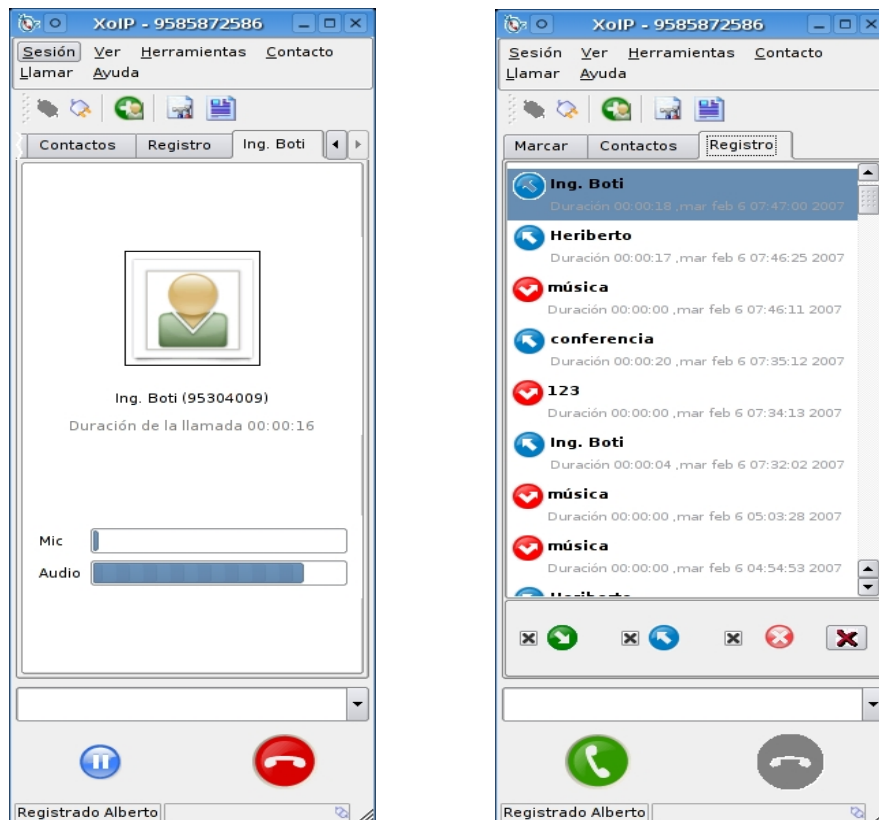


Figura A.13. a) Ventana del establecimiento de llamadas, b) Ventana de registro de llamadas.



Figura A.14. Ventana de registro.

A.9. Log de la aplicación

El log de la aplicación almacena un registro en tiempo de ejecución de todos los mensajes ocurridos durante este tiempo con la finalidad de llevar un control de su funcionamiento (Figura A.14).

A.10. Ayuda

El menú Ayuda cuenta con dos opciones:

- Manual del usuario: proporciona información sobre la utilización y las funciones del *softphone XoIP*.
- Acerca de: brinda información sobre el proyecto del *softphone XoIP* (Figura A.15)



Figura A.15. Ventana del modulo Acerca de.

B. Introducción a Asterisk

Asterisk es un conmutador telefónico privado en software (PBX) destinado a conectar redes telefónicas privadas y públicas a través de Internet. Asterisk es un software de código abierto gratuito que se ejecuta bajo las plataformas Linux, BSD, MS Windows y OS X, donde Linux es la plataforma nativa. Además de proporcionar las funciones de un Servidor/IP PBX, brinda soporte a los protocolos de establecimiento de llamada IAX, SIP, H.323, MGCP y Skinny, así como a los codecs G.729, GSM, ILBC, Speech, G.722, G.723, G.711u y G.711a.

Asterisk permite conectar teléfonos para hacer llamadas entre sí e incluso conectarse a un proveedor de VoIP o a la PSTN; inicialmente fue desarrollado por Mark Spencer de la firma Digium.

Asterisk ha revolucionado el mundo de las telecomunicaciones gracias a la idea de transportar voz sobre la red de datos; actualmente soporta el transporte de video, permitiendo aplicaciones de videollamadas y videoconferencias, además brinda diversas características como son buzón de voz, conferencias, respuesta de voz interactiva (IVR), distribución automática de llamadas, cola de llamadas, conferencias de llamada, etc., las cuales sólo eran proporcionadas por PBXs propietarios. Los usuarios de Asterisk pueden agregar nuevas funcionalidades mediante programación en el lenguaje de *script* propio de Asterisk o agregar nuevos módulos en lenguaje C o cualquier otro lenguaje de programación soportado por Linux.

Asterisk soporta diferentes tipos de hardware comercial para telefonía IP, donde el principal fabricante es Digium. Asterisk ofrece servicios de telefonía y lo siguiente como valor agregado:

- Conectar empleados desde el hogar al Servidor/IP PBX de la oficina mediante conexiones de banda ancha.
- Conectar oficinas en distintas ciudades mediante Internet o una red privada IP.
- Ofrecer buzón de voz (*voicemail*) a todos los empleados.
- Construir aplicaciones de voz interactivas.
- Ofrecer acceso al Servidor/IP PBX de la compañía a los empleados que viajan por negocios, conectándose por VPN desde un aeropuerto o *hotspots* WLAN (*Wireless Local Area Network*) en un hotel.

Asterisk ofrece las características de los sistemas de mensajería unificados de última generación, como:

- Música en espera (*musiconhold*): para los clientes que esperan en cola, soportando *streaming* así como música en formato MP3.

- Integración de sistemas *Text-to-speech*: Software de síntesis de la voz, como “Festival” o “Cepstral” pueden ser integrados.
- Generación de datos de llamada (CDR, *Call Detail Record*): Para la integración con los sistemas de facturación.
- Integración con sistemas de reconocimiento de voz: Tales como el software abierto de reconocimiento de voz *Sphinx*.
- La capacidad de interconectarse con las líneas telefónicas normales, ISDN de tarifa básica y las interfaces PRI.

B.1. Canales de comunicación

Los teléfonos analógicos, digitales y *softphones* se conectan con un canal, y algunos de ellos se registran para informar que están en línea. Los canales también se registran para las conexiones de salida a otro servidor de VoIP, por ejemplo mediante SIP a la red de *Free World Dialup* o a los proveedores de SIP como Nufone, Vonage, Siphone o IAX a *IAXtel.com*.

Asterisk soporta canales SIP, IAX y H.323, de los cuales sólo se analizan los dos primeros (sip.conf e iax.conf) debido a que el presente trabajo de tesis no abarca la explicación funcional y detallada de canales H.323.

B.2. Funciones dialplan

Para conectar las llamadas entrantes con las conexiones de salida o a otros usuarios locales Asterisk utiliza varias funciones de *dialplan*, desde funciones de lógica simple como *goto* a funciones más complejas como llamadas de *voicemail* y de conferencia.

A continuación se muestra una lista de las funciones que se pueden utilizar en el *dialplan* (*extensions.conf*). Se puede obtener una lista de aplicaciones disponibles en el CLI mediante la orden `show applications` y `show application <nombre>`.

B.2.1. Funciones generales

- *Authenticate*: Autentifica a un usuario.
- *VMAuthenticate*: Autentifica a un usuario basado en el archivo *voicemail.conf*.
- *Curl*: Permite obtener URLs externos y soporta *post* (ha sido reemplazado por CURL).
- *DUNDiLookup*: Busca un número usando DUNDi.
- *Page*: Envía un mensaje *pager* (utilizada en la versión 1.2 de Asterisk).
- *SendDTMF*: Envía dígitos DTMF.
- *SendImage*: Envía un archivo de imagen.
- *SendText*: Envía un mensaje de texto a un cliente.
- *SendURL*: Envía una URL para mostrar a un cliente.
- *System*: Ejecuta una orden del sistema.
- *Transfer*: Transfiere al usuario que llama a un interno remoto.
- *TrySystem*: Ejecutar una orden del sistema y siempre regresa un 0.
- *Wait*: Espera un cierto tiempo.
- *WaitExten*: Espera un cierto tiempo.
- *WaitForRing*: Espera a la aplicación *Ring*.

- *WaitMusicOnHold*: Pone música en espera.

B.2.2. Funciones de facturación

- *ForkCDR*: Dividir los CDR en 2 partes.
- *NoCDR*: Se asegura de que Asterisk no almacene un CDR para ciertas llamadas.
- *ResetCDR*: Reinicia los datos de los CDR.
- *SetAccount*: Fija el código de la cuenta.
- *SetAMAflags*: Fija la bandera para la facturación del canal.
- *SetCDRUserField*: Fija el campo de usuario en el CDR.
- *AppendCDRUserField*: Añade los datos al campo de usuario del CDR.

B.2.3. Funciones de administración de llamadas

- *Answer*: Contestar a un canal si suena.
- *Busy*: Indicar la condición de ocupado y esperar el corte.
- *ChanIsAvail*: Comprueba si el canal está disponible.
- *Congestion*: Indica congestión y esperar el corte.
- *Dial*: Establecer una llamada y conectarla con el canal actual.
- *DISA*: Acceso directo al sistema interno.
- *Hangup*: Corte incondicional.
- *RetryDial*: Hacer una llamada, reintentando en caso de falla, permitiendo un interno opcional de salida.
- *Ringin*: Indicar el tono de llamada.

B.2.4. Funciones de identificador de llamada

- *CallingPres*: Cambiar la presentación para el *callerid*.
- *LookupBlacklist*: Buscar el nombre/número de *callerid* en la lista negra de una base de datos.
- *LookupCIDName*: Buscar el nombre del *callerid* de una base de datos local.
- *PrivacyManager*: Requerir que se ingrese el número de teléfono si no se envió ningún *callerid*.
- *SetCallerID*: Concede el *callerid* (ha sido reemplazada por CALLERID).
- *SetCallerPres*: Ajuste independiente de canal para la presentación del que llama.
- *SetCIDName*: Fija el nombre de *callerid* (ha sido reemplazado por CALLERID).
- *SetCIDNum*: Fija solamente el número de identificación del que llama (no el nombre) (ha sido reemplazado por CALLERID).
- *SoftHangup*: Hace un *hangup* en otro canal
- *Zapateller*: Bloquea a los *telemarketers*.

B.2.5. Funciones ADSI

ADSIProg: Cargar scripts ADSI de Asterisk en un teléfono.

GetCPEID: Obtener la identificación de CPE ADSI.

B.2.6. Funciones de manejo de bases de datos

- *DBdel*: Suprimir un valor de la base de datos.
- *DBdeltree*: Suprimir una familia o un árbol de valores de la base de datos.
- *DBget*: Obtener un valor de la base de datos (ha sido reemplazado por la función DB).
- *DBput*: Almacenar un valor en la base de datos (ha sido reemplazado por la función DB).
- *MYSQL*: Realiza varias actividades en la base de datos de MySQL.
- *DBQuery*: Ejecuta sentencias predefinidas en servidores de MySQL, y envía los resultados nuevamente al *dialplan*.

B.2.7. Funciones para la integración de aplicaciones

- *AGI*: Ejecuta una aplicación de AGI.
- *DeadAGI*: Ejecuta un AGI en un canal colgado.
- *EAGI*: Ejecuta una aplicación de AGI.
- *EnumLookup*: Busca un número en ENUM.
- *ExternalIVR*: Ejecuta un generador de IVR externo.
- *Macro*: Implementación de macros.
- *NoOp*: Ninguna operación, puede imprimir valores en la consola *debug*.
- *Perl*: El *res_perl* en Asterisk es el equivalente al *mod_perl* de Apache.
- *PHP*: Integra PHP en Asterisk sin necesidad de AGI.
- *Read*: Lee una variable con DTMF.
- *TXTCIDName*: Busca el nombre del que llama de un registro TXT
- *UserEvent*: Enviar un acontecimiento arbitrario a la interfaz del administrador.

B.2.8. Funciones para el control de flujos y tiempos

- *AbsoluteTimeout*: Brinda el tiempo máximo absoluto para una determinada llamada.
- *DigitTimeout*: Fija el intervalo máximo entre los dígitos.
- *Gosub*: Saltar a una subrutina y volver.
- *GosubIf*: Salto condicional a una subrutina y vuelta.
- *Goto*: Ir a una prioridad particular, un interno o un contexto particular.
- *GotoIf*: Función *goto* condicional.
- *GotoIfTime*: Función *goto* condicional en el tiempo actual.
- *Random*: Hacer un salto al azar en el *dialplan*.
- *ResponseTimeout*: Brinda el máximo tiempo de espera de respuesta
- *Return*: Retorno de un *gosub* o de un *gosubIf*.
- *StackPop*: Borra una dirección de retorno sin volver.
- *While*: Comienza un ciclo *while*.
- *EndWhile*: Terminar un ciclo *while*.
- *ExecIf*: *Exec* condicional.

B.2.9. Funciones para la manipulación de variables y strings

- *ImportVar*: Determina o fija el valor de una variable.
- *Math*: Realiza cálculos simples (ha sido reemplazada por la función MATH).
- *SetGlobalVar*: Determina el valor de una variable (ha sido reemplazada por la función GLOBAL).
- *Set*: Fija las variables del canal o los valores de las funciones.
- *DBRewrite*: Ejecuta expresiones regulares de perl y substituciones desde una base de datos de MySQL.

B.2.10. Funciones de sonido PlayBack

- *Background*: Coloca un sonido mientras se ejecutan otras órdenes.
- *BackgroundDetect*: Pone un sonido de fondo y activa la detección de habla.
- *ControlPlayback*: Pone un sonido con controles de rebobinado, adelanto y de salida.
- *DateTime*: Decir la fecha y/o el tiempo (obsoleto).
- *Echo*: Activa un eco de lo que dice el usuario
- *Festival*: Decir un texto con el sintetizador de voz “Festival”.
- *Flite*: Decir un texto con el sintetizador de voz “Festival Lite” (tiene una respuesta más rápida que el “Festival”).
- *Milliwatt*: Genera un tono constante de 1000 Hz en 0 dBm (ulaw).
- *MP3Player*: Pone un sonido MP3.
- *MusicOnHold*: Pone música en espera indefinidamente.
- *Playback*: Pone un sonido.
- *Playtones*: Pone una lista de tonos mientras que ejecuta otras órdenes.
- *Progress*: Poner audio al que llama antes de contestar la línea.
- *SayUnixTime*: Decir la fecha y/o tiempo.
- *SayAlpha*: Dice Alfa.
- *SayDigits*: Dice dígitos.
- *SayNumber*: Dice números.
- *SayPhonetic*: Dice fonética.
- *SetMusicOnHold*: Determina la clase de música en espera por defecto.
- *SetLanguage*: Cambia el lenguaje para los archivos de sonido.
- *StopPlaytones*: Deja de poner una lista de tonos.

B.2.11. Funciones de sonido grabación y monitoreo

- *ALSAMonitor*: Monitorear la consola de ALSA.
- *ChangeMonitor*: Cambiar el nombre del fichero de monitoreo de un canal.
- *ChanSpy*: Escucha una llamada.
- *Dictate*: Grabar y reproducir un dictado.
- *MixMonitor*: Graba y mezcla los 2 canales de una llamada nativamente.

- *Monitor*: Grabar una conversación de teléfono a un archivo de sonido.
- *Record*: Grabar la entrada de voz de un usuario a un archivo.
- *StopMonitor*: Deja de monitorear un canal.

B.2.12. Funciones SIP

- *SIPdtmfMode*: Cambia el modo DTMF de una llamada SIP.
- *SIPGetHeader*: Obtiene cualquier encabezado de un mensaje SIP INVITE (substituido por SIP_HEADER()).
- *SIPAddHeader*: Agrega un encabezado a un mensaje SIP INVITE de salida.

B.2.13. Funciones ZAP

- *Flash*: Envía un Flash en un Trunk Zap.
- *ZapBarge*: Escucha en un canal Zap.
- *ZapSendKeypadFacility*: Envía dígitos fuera de banda sobre una PRI.
- *ZapRAS*: Proporcionar un servicio de datos ISDN.
- *ZapScan*: Explorar los canales Zap para monitorear llamadas.

B.2.14. Funciones para voicemail y conferencia

- *Directory*: Proporcionar un directorio de internos del *voicemail*.
- *HasNewVoicemail*: Si hay nuevo *voicemail* salta a la prioridad +101.
- *MailboxExists*: Comprueba si existe la casilla.
- *MeetMe*: Conferencia simple *MeetMe*.
- *MeetMeAdmin*: Administración de la conferencia de *MeetMe*.
- *MeetMeCount*: Conteo de participantes de una conferencia *MeetMe*.
- *VoiceMail*: Dejar un mensaje de *voicemail*.
- *VoiceMailMain*: Ingresar al sistema de *voicemail*.
- *VMAuthenticate*: Autenticar un usuario basado en el archivo *voicemail.conf*.

B.2.15. Funciones para colas de espera y administración de agentes

- *AddQueueMember*: Agrega dinámicamente miembros de la cola de espera.
- *AgentCallbackLogin*: *Callback* a un agente.
- *AgentLogin*: Pide al agente que se registre.
- *AgentMonitorOutgoing*: Graba la llamada saliente del agente.
- *ParkAndAnnounce*: Estacionar una llamada y anunciarla.
- *ParkedCall*: Contestar una llamada estacionada.
- *PauseQueueMember*: Pone en pausa a un agente.
- *Queue*: Pone una llamada entrante en una cola de espera.
- *RemoveQueueMember*: Quita dinámicamente a miembros de una cola de espera.
- *UnpauseQueueMember*: Rehabilita a un agente.

B.2.16. Funciones de alarma de monitoreo

- *AlarmReceiver*: Emular el protocolo *Contact ID* receptor de alarmas de *Ademco*.

B.2.17. Funciones radio amateur

- *Rpt*: Soporte para conectar radio aficionados y repetidores comerciales de dos vías.

B.2.18. Funciones externas

- *App_dbodbc*: Modificadores del *Dialplan* usando unixODBC.
- *DynExtenDB*: Almacenar los pares internos en una base de datos.
- *Iconv*: Convertir conjuntos (*sets*) de caracteres.
- *LDAPget*: Obtener un valor de un servidor de directorio LDAP.
- *CallingCard*: Diseñada para *Postgres*.
- *PPPD*: Conector al demonio de PPP.
- *Backticks*: Almacenar el resultado de una orden del *shell* en una variable de Asterisk.
- *ASR*: Reconocimiento de voz multilingüe profesional para el Asterisk.

B.2.19. Aplicaciones bristuff y zaphfc

- *PickUp*: Atender un canal.
- *PickUpChan*: Tomar el canal especificado.
- *PickDown*: Colgar una llamada que suena.
- *Steal*: Asumir el control de una llamada establecida.
- *Devstate*: Generar un evento de cambio de estado del dispositivo (en uso, ocupado, sonando).
- *Segfault*: Cerrar Asterisk con un segfault.
- *ZapEC*: Permitir o anular la cancelación de eco para Zap.

B.2.20. Funciones para los canales Sirrix

- *SrxEchoCan*: Inhabilitar/permitir la cancelación de eco.
- *SrxDeflect*: Desviar una llamada entrante.
- *SrxMWI*: Determinar/reiniciar MWI (*MessageWaiting Indication*) en un grupo de Sirrix.

El *dialplan* se almacena en el archivo de configuración *extensions.conf*, el cual se conectan las acciones con los pares internos. Cada par interno pertenece a un contexto, el contexto por defecto o un contexto específico: como llamadas entrantes por SIP, llamadas de salida interurbanas PSTN, llamadas locales, llamadas entre oficinas, etc. Los usuarios que se conectan a Asterisk pertenecen todos a un contexto, especificado en el archivo de configuración del canal, que es donde Asterisk busca cómo manejar las llamadas realizadas por ese usuario y comprueba los permisos de acceso a las líneas costosas, con reglas distintas para usuarios locales y contactos que llaman de una línea exterior.

En el *dialplan* se determinan todas las acciones y situaciones que el Servidor/IP PBX debe manejar; se pueden configurar contextos que trabajen durante parte del día o de la noche, se pueden incluir contextos de otro contexto y simplificar o hacer un *dialplan* muy complicado.

Ejemplos de lo que se puede realizar en un *dialplan*:

- Conectar una llamada con el *voicemail* si un usuario no contesta al teléfono en un plazo de 20 segundos.
- Conectar una llamada a una conferencia multipartita.
- Transferir las llamadas a otro Asterisk.
- Bloquear las llamadas de un origen no identificado o indeseado.
- Buscar datos en una base de datos basándose en el *callerId* y decidir qué grupo de agentes deberán contestar la llamada.
- Crear colas de llamada y dejar que un grupo de agentes maneje las llamadas entrantes.

B.3. Archivos de configuración de Asterisk

Asterisk utiliza los siguientes archivos para su correcto funcionamiento:

- *Adsi.conf*: Contiene la configuración para la interfaz analógica de servicios por pantalla (ADSI, *Analogue Display Service Interface*).
- *Adtranvoifr.conf*: Contiene la configuración para usar voz sobre *Frame Relay* de *adtran*.
- *Agents.conf*: Este archivo tiene la configuración para usar agentes; como en una central telefónica, éste permite definir agentes y asignarles su respectivos IDs y contraseñas.
- *Alarmreceiver.conf*: Configura la aplicación receptora de alarmas.
- *Alsa.conf*: Este archivo contiene las variables de configuración para la tarjeta de sonido de la consola.
- *Asterisk.ads*: Contiene el *script* ADSI por defecto de Asterisk. Es ejecutado por el teléfono si se utiliza hardware ADSI.
- *Asterisk.conf*: Este archivo fija ciertas variables para el uso de Asterisk, muchas de las cuales no se requiere modificar.
- *Cdr_manager.conf*: Este archivo configura los registros de detalle de llamada (CDR, *Call Detail Records*) para su administración.
- *Cdr_odbc.conf*: Archivo de configuración para usar una conexión de base de datos (ODBC, *Open DataBase Connectivity*) con la finalidad de almacenar los CDR's.
- *Cdr_pgsq*.*conf*: Permite usar una base de datos PostgreSQL para almacenar los CDR's.
- *Cdr_tds.conf*: Este es el archivo de configuración para usar FreeTDS, permitiendo conexiones a *Microsoft SQL* y *Sybase*.
- *Enum.conf*: Este archivo configura el uso de ENUM, el cual permite resolver números de teléfono sobre DNS, permitiendo encaminar las llamadas por IP en lugar de pasarlas por PSTN.
- *Extconfig.conf*: Permite elegir cargar las colas vía el motor de base de datos.
- *Extensions.conf*: Este archivo configura el comportamiento de Asterisk.
- *Features.conf*: Este archivo contiene opciones para el *parking* de llamadas, así como algunas características misceláneas.
- *Festival.conf*: Este archivo fija los parámetros para "Festival", el cual es un programa de código abierto que permite leer texto al servidor.
- *Iax.conf*: Este archivo configura conversaciones de VoIP usando el protocolo IAX.
- *Iaxprov.conf*: Permite simple aprovisionamiento de S1011 de Digium (también conocido como IAXy).

- *Indications.conf*: Configura ciertos comportamientos del sistema telefónico, como pueden ser cadencias de *rings* y tonos.
- *Logger.conf*: Establece el tipo de registro que se usará, generalmente se utiliza la configuración por defecto.
- *Manager.conf*: Configura el acceso remoto al administrador de llamadas de Asterisk.
- *Meetme.conf*: Este archivo de configuración establece salas de conferencia.
- *Mgcp.conf*: Este archivo configura el protocolo MGCP.
- *Modem.conf*: Este archivo fija ciertas variables para la utilización de determinados módems con Asterisk.
- *Modules.conf*: Este archivo de configuración selecciona los módulos de Asterisk que serán activados.
- *Musiconhold.conf*: Este archivo de configuración crea instancias de música en espera (MOH, *Music On Hold*) y define que música se reproducirá. Hasta el momento sólo soporta la reproducción de MP3 si se tiene instalado *mpg123*.
- *Osp.conf*: Permite configurar el subsistema del protocolo de establecimiento abierto (OSP, *Open Settlement Protocol*) de Asterisk.
- *Oss.conf*: Este archivo de configuración es parecido a *alsa.conf*.
- *Phone.conf*: Este archivo permite el uso de algunas interfaces de telefonía Linux, por ejemplo *linejack* por *quicknet* o hardware Digium.
- *Privacy.conf*: Permite configurar las opciones de privacidad.
- *Queues.conf*: Permite crear colas para las personas que llaman (*callers*). Las colas son una función muy utilizada ya que permiten proporcionar servicios más profesionales a los clientes; gracias a las colas, el sistema es capaz de responder inmediatamente a las llamadas sin considerar si hay disponibilidad o no de operadores. Si todas las líneas están ocupadas o no disponibles, la llamada será retenida en una cola hasta que sea respondida por el próximo agente de llamada disponible.
- *Res_config_odbc.conf*: Este archivo fija la configuración para almacenar nuestras opciones de configuración en un ODBC de base de datos.
- *Res_odbc.conf*: Esta es otra pieza de configuración para almacenar las opciones de configuración en un ODBC de base de datos.
- *Rpt.conf*: Permite usar un repetidor de radio.
- *Rtp.conf*: Fija los puertos a usar por el protocolo de tiempo real.
- *Sip.conf*: Este archivo de configuración define el protocolo de inicio de sesión (SIP), los usuarios y sus opciones.
- *Skinny.conf*: Este archivo configura el protocolo *Skinny* para VoIP, el cual es utilizado por muchos de los teléfonos Cisco.
- *Telecordia-1.ads*: Este es otro ejemplo de *script* ADSI.
- *Voicemail.conf*: Este archivo de configuración crea usuarios *voicemail* y algunas opciones globales para *comedian* mail (sistema *voicemail* de Asterisk).
- *Vpb.conf*: Este archivo configura hardware *voicetronix*.
- *Zapata.conf*: Este archivo especifica las opciones de configuración de las interfaces de telefonía zapata.

Los archivos de configuración que se utilizaron en el presente trabajo de tesis son `extensions.conf`, `iax.conf`, `meetme.conf`, `musiconhold.conf`, `sip.conf`, `voicemail.conf` y `zapata.conf`. Los cuales son suficientes para trabajar con VoIP dentro de una LAN o una WAN como Internet. Dado que no es objetivo de la tesis hablar ampliamente sobre *Asterisk*, sólo se muestran los archivos utilizados.

B.3.1. Asterisk.conf

Este archivo indica a Asterisk la localización de los módulos necesarios; dicho archivo se crea automáticamente y no se recomienda modificarlo. A continuación se muestra el contenido del archivo utilizado en el presente proyecto:

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run
astlogdir => /var/log/asterisk
```

B.3.2. Extensions.conf

Este archivo es uno de los más utilizados e importantes dentro los archivos de configuración de Asterisk ya que define cómo se manejarán las llamadas de entrada y de salida. Contiene todos los números de extensiones y está dividido por secciones llamadas contextos, donde cada contexto define a una o más extensiones.

```
;contexto general
[general]
static=yes      ; estas dos líneas previenen que desde la línea de órdenes
writeprotect=yes ; se pueda sobrescribir el archivo de configuración
; Contexto que define a todas las extensiones SIP
[telsip]
exten=>95304008,1,Dial(SIP/95304008,30,Ttr)
exten=>95304008,2,Voicemail(u95304008)
exten=>95304008,102,Voicemail(b95304008)
exten=>95304008,103,Hangup

exten=>95304009,1,Dial(SIP/95304009,30,Ttr)
exten=>95304009,2,Voicemail(u95304009)
exten=>95304009,102,Voicemail(b95304009)
exten=>95304009,103,Hangup
; Se define un número donde los usuarios puedan alcanzar el buzón de voz y se pueda llamar a la
; aplicación voicemailmain, pasándole como variable argumento, el número del que se llama
; Así que lo único que se tecleará es la contraseña
exten=>*86,1,VoicemailMain(${CALLERIDNUM})

; Se define una extensión para escuchar el musiconhold
exten=>40,1,WaitMusicOnHold(30)
```

```
; Se incluye la ayuda y los salones para conferencia
include=>ayuda
include=>meetme
; Se define que los clientes SIP, puedan llamar a clientes IAX
include=>teliarx
; Extensión creada para ver el tiempo de respuesta con el Servidor/IP PBX
include=>echoprueba
; Contexto que define a todas las extensiones IAX
[teliarx]
exten=>9531021271,1,Dial(IAX2/9531021271,30,Ttr)
exten=>9531021271,2,Voicemail(u9531021271)
exten=>9531021271,102,Voicemail(b9531021271)
exten=>9531021271,103,Hangup

exten=>9585872586,1,Dial(IAX2/9585872586,30,Ttr)
exten=>9585872586,2,Voicemail(u9585872586)
exten=>9585872586,102,Voicemail(b9585872586)
exten=>9585872586,103,Hangup

; Se define que clientes IAX puedan llamar a clientes SIP
include=>telsip
; Contexto de ayuda
[ayuda]
exten=>111,1,Wait,1           ; Espera un segundo
exten=>111,2,Answer          ; Responde la línea
exten=>111,3,DigitTimeout,5   ; Determina el tiempo de digitar en 5 segundos
exten=>111,4,ResponseTimeout,10 ; Determina el tiempo de respuesta en 10 segundos
exten=>111,5,BackGround(demo-congrats) ; Reproduce el mensaje de felicitaciones
exten=>111,6,BackGround(demo-instruct) ; Reproduce algunas instrucciones

exten=>222,1,BackGround(demo-moreinfo) ; Proporciona más información
exten=>222,2,Goto(111,6)

exten=>333,1,Playback(demo-abouttotry) ; Dice qué está pasando
exten=>333,2,Dial(IAX2/guest@misery.digium.com/s@default)
exten=>333,3,Playback(demo-nogo)      ; No pudo conectar el demo
exten=>333,4,Goto(111,6)

; Contexto de la sala de conferencia
[meetme]
exten=>200,1,Answer
exten=>200,2,Playtones,ring
exten=>200,3,Wait,3
exten=>200,4,Meetme(4000|M)
exten=>200,5,Hangup
```

```

; Contexto de prueba
[echoprueba]
exten=>*46,1,Answer()
exten=>*46,2,Playback(welcome)
exten=>*46,3,Playback(demo-echotest)
exten=>*46,4,Echo()
exten=>*46,5,Playback(demo-echodone)
exten=>*46,6,Playback(vm-goodbye)
exten=>*46,7,Hangup

```

B.3.3. iax.conf

En este archivo se definen los clientes que hacen uso del protocolo IAX y que podrán registrarse en el Servidor/IP PBX.

```

; Archivo que permite el registro de usuarios IAX
; Contexto general
[general]
bindaddr=0.0.0.0 ; Permite registrarse desde cualquier dirección IP de la red.
bandwidth=low ; Utilizar ancho de banda bajo
disallow=all ; Desactiva todos los codecs
allow=gsm ; Permite el codec gsm
allow=ilbc ; Permite el codec ilbc
allow=ulaw ; Permite el codec ulaw
allow=alaw ; Permite el codec alaw
allow=speex ; Permite el codec speex
jitterbuffer=no ; Se refiere a la variación de la latencia entre paquetes
forcejitterbuffer=no; Forzar el jitterbuffer
tos=throughput ; Define el tipo de servicio, en este caso maximiza el throughput
language=es ; Se define el lenguaje en este caso: español
; Cliente 9531021271
[9531021271]
type=friend ; Permite realizar y recibir llamadas
username=9531021271 ; Nombre de usuario
secret=beto ; Contraseña
auth=md5 ; Método de autenticación requerido
qualify=yes ; Minimiza la cantidad de overhead
host=dynamic ; Host al que se conectaran los clientes
context=teliax ; Contexto en el que se encuentra en extensions.conf
mailbox=106 ; Número de buzón de voz
callerid= "Administrador"<9531021271> ; identificador de llamada
disallow=all ; Desactiva todos los codecs
allow=gsm ; Activa el codec gsm para este par
;allow=ilbc
;allow=alaw
;allow=ulaw
;allow=speex
; Cliente 9585872586

```



```
[9585872586]
type=friend
username=9585872586
secret=200480
auth=md5
qualify=yes
host=dynamic
context=teliax
mailbox=107
callerid= "Alberto"<9585872586>
disallow=all
;allow=gsm
allow=ilbc
;allow=ulaw
;allow=alaw
;allow=speex
```

B.3.4. meetme.conf

Este archivo define las salas de conferencia.

```
[rooms]
conf=>4000 ; Se crea la sala de conferencia 4000 con acceso libre a los usuarios registrados
```

B.3.5. musiconhold.conf

Archivo que define el directorio en que se encuentran los archivos de música y la forma en que se reproducirán.

```
;musiconhold.conf
[classes]
; Directorio donde se localizan los MP3, el -Z es
; para que los toque de manera aleatoria
; y desde donde se quedó la ultima vez la pista
default=>quietmp3:/var/lib/asterisk/mohmp3,-Z
```

B.3.6. sip.conf

Archivo en el cual se definen los clientes que hacen uso del protocolo SIP y que podrán registrarse en el Servidor/IP PBX.

```
[general]
port=5060 ; Se define el puerto que utilizan los teléfonos con tecnología SIP (5060)
bindaddr=0.0.0.0 ; Direcciones IP que se usarán (todas las que existan)
context=telsip ; Envía las llamadas SIP que no conocemos al contexto telsip de extensions.conf
tos=lowdelay ; Tipo de servicio de bajo retardo
dtmfmode=rfc2833 ; Tonos dtmf
disallow=all ; Deshabilita todos los codecs
allow=ulaw ; Permite el codec ulaw
allow=alaw ; Permite el codec alaw
allow=speex ; Permite el codec speex
```

```

allow=ilbc      ; Permite el codec ilbc
allow=gsm       ; Permite el codec gsm
;allow=g729     ; Permite el codec g729 2.5 kb/s (codec propietario)
language=es     ; Lenguaje utilizado
;Cliente SIP 95304008
[95304008]
type=friend     ; Realiza y recibe llamadas
secret=beto     ; Contraseña definida, el nombre de usuario es el mismo que el contexto
callerid="Heriberto" <95304008> ; Identificador de llamada
host=dynamic    ; Host utilizado
context=telsip  ; Contexto en el que se encuentra definido en el archivo extensions.conf
canreinvite=yes ; Permite conectar directamente los pares
dtmfmode=rfc2833; Uso de DTMF
transfer=yes    ; Permite transferencia
nat=yes        ; Traducción de direcciones de red
disallow=all    ; Deshabilita todos los codecs
allow=speex     ; Permite el codec speex
mailbox=100     ; Número de buzón de voz utilizado
;Cliente SIP 95304009
[95304009]
type=friend
secret=toño
callerid="Toño TallerElectronica" <95304009>
host=dynamic
context=telsip
canreinvite=yes
dtmfmode=rfc2833
transfer=yes
nat=yes
disallow=all
allow=ulaw
mailbox=101

```

B.3.7. voicemail.conf

Archivo de configuración en el cual se definen parámetros del buzón de voz.

```

[general]
; Dirección de correo que aparece en las notificaciones
serveremail=voiputm@voiputm.homeip.net
; Nombre del destinatario con el que aparece el e-mail
fromstring= Tesis (Alberto Lavariega Arista)
; El e-mail que recibirá el usuario va a tener el voicemail como attachment?
attach=no
; Formato en que debe escribirse el voicemail
format=wav49|gsm|wav
skipms=3000
maxsilence=10

```

```
silencethreshold=128
maxlogins=3
; Cuerpo del mensaje del e-mail que se enviará
emailbody= Estimado ${VM_NAME}:\n\n\t La persona con número de teléfono ${VM_CALLERID} te
ha dejado un correo de voz de ${VM_DUR} de duración (numero ${VM_MSGNUM})\n en tu buzón
${VM_MAILBOX} el día ${VM_DATE}, así que te sugerimos revisar tu bandeja de buzón de mensajes de
voz marcando 86 y posteriormente tu contraseña.\n\n\n Saludos.\n\nATENTAMENTE\nConmutador: voi-
putm.homeip.net\n

[zonemessages]
eastern=America/NewYork|'vm-received' Q 'digits/at' IMp
central=America/Chicago|'vm-received' Q 'digits/at' IMp
central24=America/Chicago|'vm-received' q 'digits/at' H 'digits/hundred' M 'digits/hours'

[default]
; Aquí se declaran los buzones de voz, sus claves y los correos a donde se enviará la notificación
95304008=>008,Alberto Lavariega,huatulco_beto@hotmail.com
95304009=>009, Heriberto,herhdez@hotmail.com
9531021271=>271,Alberto Lavariega,huatulco_beto@hotmail.com
9585872586=>86,Ala,aristalbertolea@hotmail.com
```

B.3.8. Zapata.conf

Archivo utilizado para habilitar el *musiconhold*.

```
[channels]
language=es
context=default
usecallerid=yes
hidecallerid=no
callwaiting=yes
usecallingpres=yes
callwaitingcallerid=yes
threewaycalling=yes
transfer=yes
canpark=yes
cancallforward=yes
callreturn=yes
echocancel=yes
echocancelwhenbridged=yes
rxgain=0.0
txgain=0.0
group=1
callgroup=1
pickupgroup=1
immediate=no
musiconhold=default
```


C. Especificación de caso de usos

A continuación se detalla el resto de casos de uso involucrados en el presente trabajo de tesis.

C.1. Especificación de caso de usos Eliminar cuenta

El caso de uso Eliminar Cuenta permite al usuario eliminar las cuentas de usuario que ya no necesite o con la finalidad de que los parámetros de conexión (usuario y servidor) no sean visibles para cualquier otro usuario que use el mismo equipo de cómputo. En este caso de uso se tiene al usuario como único actor.

La Tabla C.1 muestra el historial de revisiones y la Tabla C.2 el flujo básico del caso de uso Eliminar cuenta.

Tabla C.1. Historial de revisiones del caso de uso Eliminar cuenta.

Fecha	Versión	Descripción	Autor
11/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
12/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A
13/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.2. Flujo básico del caso de uso Eliminar cuenta.

Actor	Sistema
1. Este caso de uso inicia cuando el usuario decide eliminar una cuenta de usuario	
2. El usuario se encuentra en la ventana de configuración y ha seleccionado la cuenta a eliminar	
3. Presiona el botón de Eliminar cuenta	4. Envía un mensaje preguntando si realmente se desea eliminar la cuenta
5. El usuario selecciona la opción Si para eliminar la cuenta	6. Elimina la cuenta seleccionada
7. Finaliza el caso de uso	

Como excepción, al momento de eliminar una cuenta de usuario, el sistema preguntará mediante un mensaje, si realmente se desea eliminar esa cuenta. Una vez eliminada la cuenta, los datos de conexión no podrán ser recuperados a menos que el usuario las recuerde.

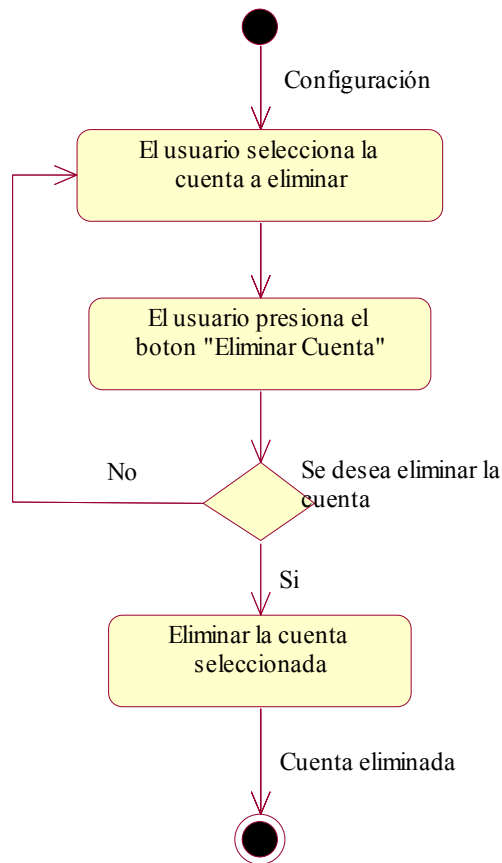


Figura C.1. Diagrama de actividades del caso de uso Eliminar cuenta.

Para que el usuario pueda ejecutar este caso de uso, antes debió haber creado una cuenta y debe estar posicionado en la ventana de configuración para proceder a eliminar la cuenta que haya seleccionado.

Las Figuras C.1-C.3 muestran los diagramas de actividades, de secuencia y de clases, respectivamente, del caso de uso Eliminar cuenta.

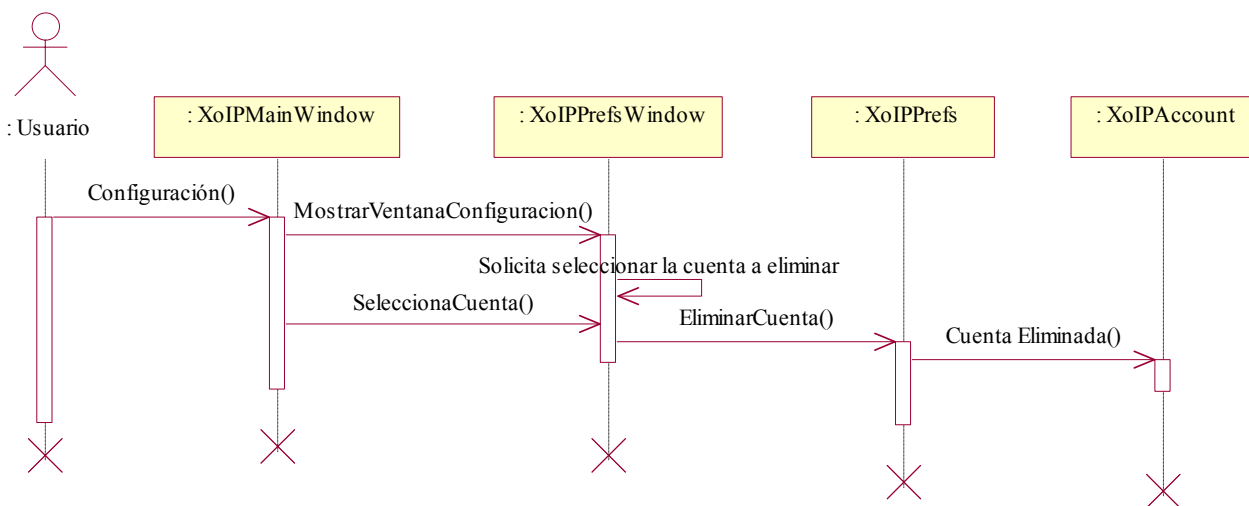


Figura C.2. Diagrama de secuencia del caso de uso Eliminar cuenta.

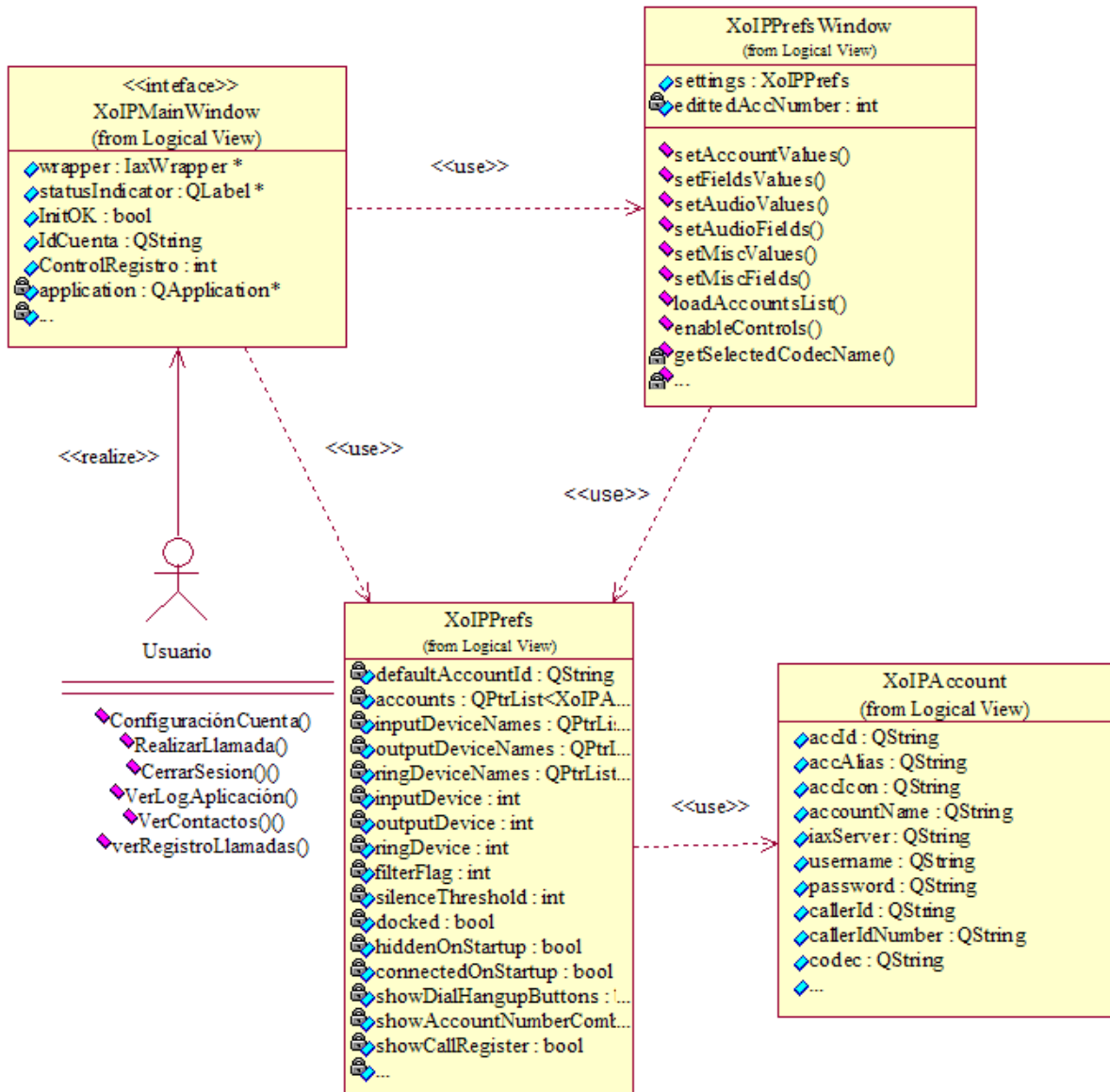


Figura C.3. Diagrama de clases del caso de uso Eliminar cuenta.

C.2. Especificación de caso de uso Modificar cuenta

El caso de uso Modificar cuenta permite al usuario modificar cuentas de usuario, ya sea para cambiar los datos del servidor de conexión, los filtros o para elegir un nuevo codec. El objetivo de este caso de uso es modificar una cuenta de usuario previamente configurada. Este caso de uso tiene al usuario como único actor.

La Tabla C.3 muestra el historial de revisiones y la Tabla C.4 el flujo básico del caso de uso Modificar cuenta.

Tabla C.3. Historial de revisiones del caso de uso Modificar cuenta.

Fecha	Versión	Descripción	Autor
14/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariéga A.
15/Marzo/2006	1.1	Terminación documento	Alberto Lavariéga A
16/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariéga A.

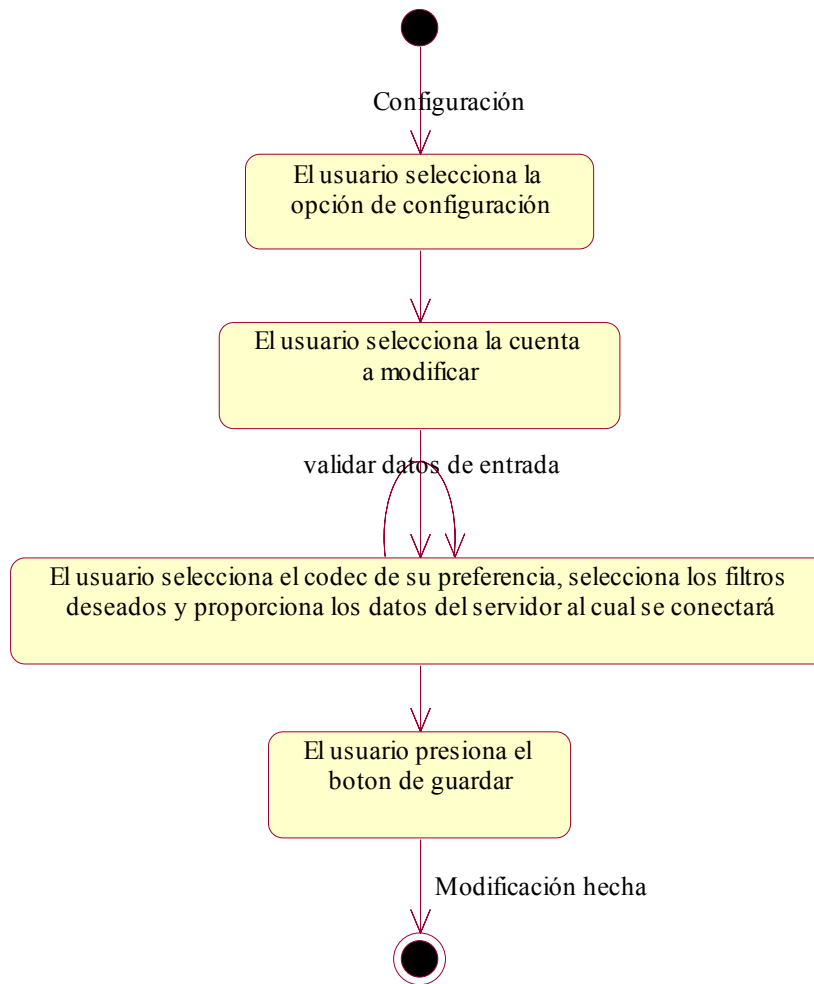
Tabla C.4. Flujo básico del caso de uso Modificar cuenta.

Actor	Sistema
1. Este caso de uso comienza cuando el usuario decide cambiar ciertas características de su cuenta	
2. El usuario se encuentra en la ventana de configuración	
3. Modifica los datos que desea	
4. Presiona el botón Guardar	5. Almacena las modificaciones realizadas
6. Finaliza el caso de uso	

Como excepción, si llegan a faltar datos para la configuración, el sistema indicará mediante un mensaje que faltan datos y por lo tanto no se almacenarán las modificaciones realizadas hasta que se hayan proporcionado dichos datos.

Para que el usuario pueda llevar a cabo este caso de uso se debe encontrar en la ventana de configuración y debió haber seleccionado una cuenta para realizarle las modificaciones correspondientes.

Las Figuras C.4-C.6 muestran los diagramas de actividades, de secuencia y de clases, respectivamente, del caso de uso Modificar cuenta.

**Figura C.4.** Diagrama de actividades del caso de uso Modificar cuenta.

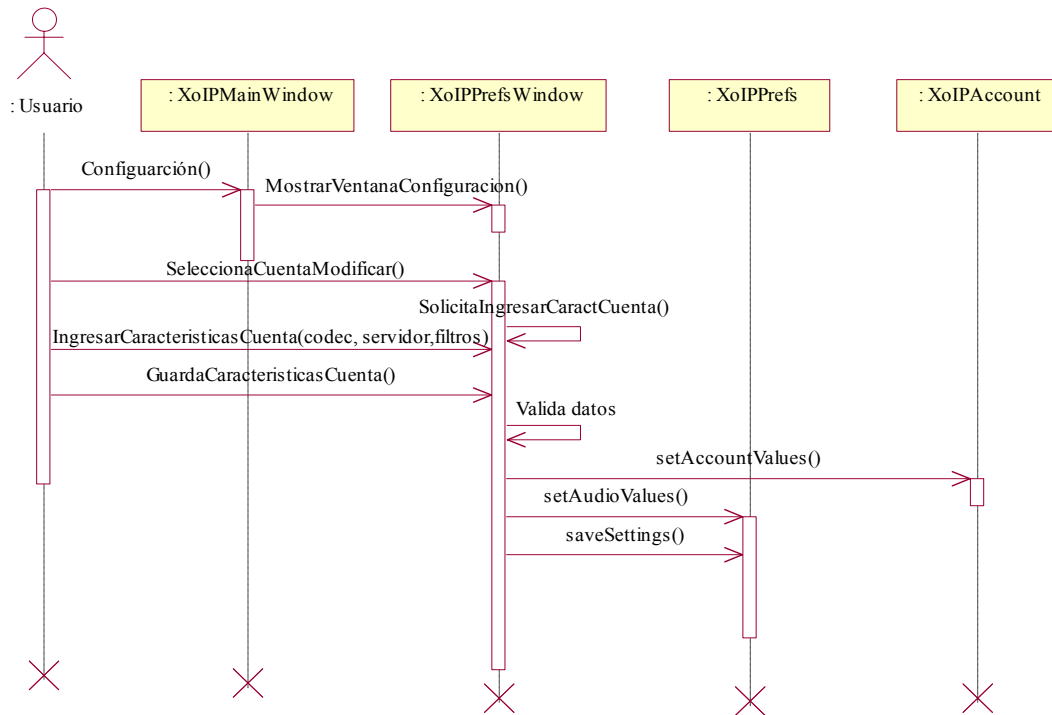


Figura C.5. Diagrama de secuencia del caso de uso Modificar cuenta.

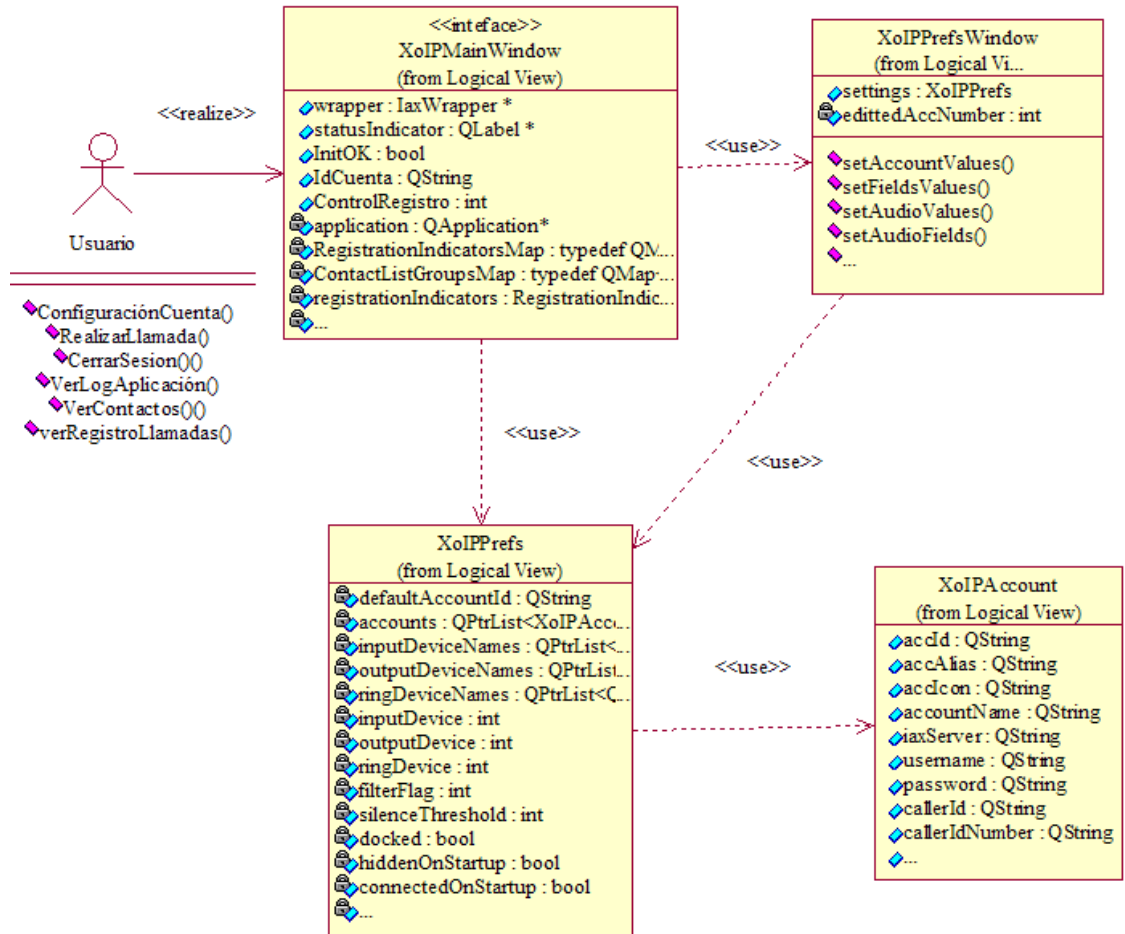


Figura C.6. Diagrama de clases del caso de uso Modificar cuenta.

C.3. Especificación de caso de uso Configurar cuenta

Este caso de uso permite al usuario asignar los filtros necesarios para asegurar calidad en las llamadas VoIP. En este caso de uso se tiene al usuario como único actor.

La Tabla C.5 muestra el historial de revisiones y la Tabla C.6 el flujo básico de eventos del caso de uso Configurar cuenta.

Tabla C.5. Historial de revisiones del caso de uso Configurar cuenta.

Fecha	Versión	Descripción	Autor
17/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
18/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A.
19/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.6. Flujo básico del caso de uso Configurar cuenta.

Actor	Sistema
1. Este caso de uso inicia cuando el usuario desea utilizar cualquiera de los siguientes filtros: reducción de ruido, supresión de silencios, eliminación de eco, enviar tramas de ruido cuando se detecta silencio, detección automática de la ganancia analógica, detección automática de la ganancia digital.	
2. El usuario da click en la opción de configuración y de ahí selecciona la pestaña Audio .	
3. Selecciona los filtros deseado	
4. Click en guardar la configuración realizada	5. Almacena la configuración
6. Finaliza el caso de uso	

Si al momento de almacenar la configuración, el sistema envía un mensaje de que hacen falta parámetros, es porque el caso de uso no ha sido completado.

Para llevar a cabo este caso de uso, el usuario debe estar en la ventana de configuración y debió haber creado una cuenta de usuario para probar la configuración realizada.

Las Figuras C.7-C.9 muestran los diagramas de secuencia, de actividades y de clases, respectivamente, del caso de uso Configurar cuenta.

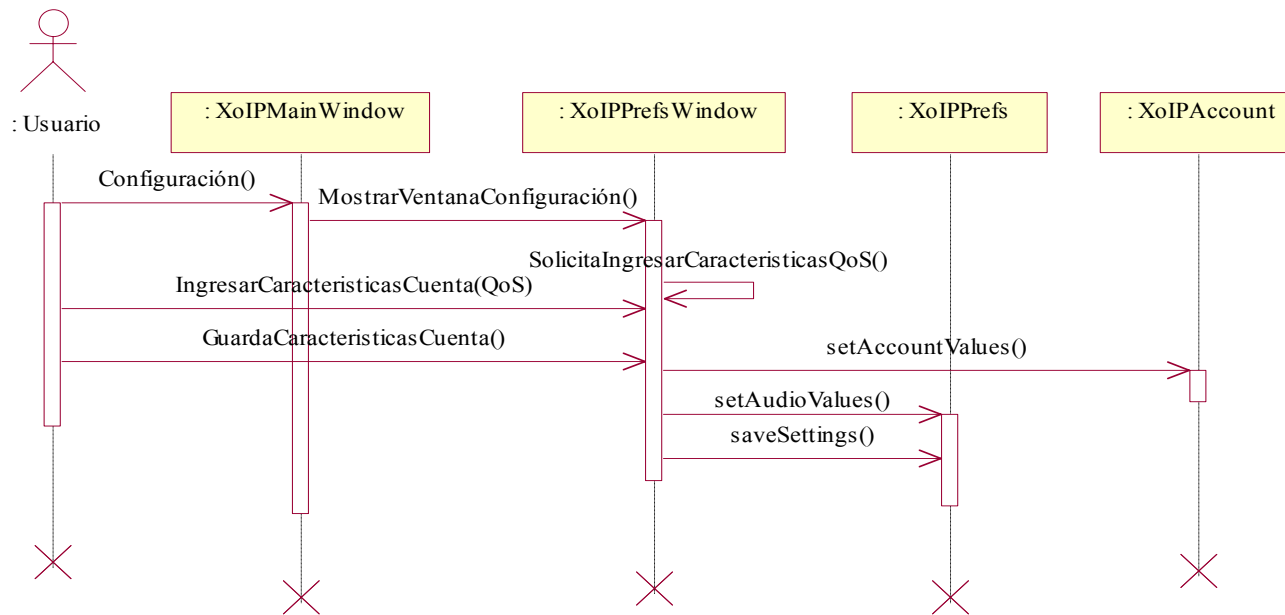


Figura C.7. Diagrama de secuencia del caso de uso Configurar cuenta.

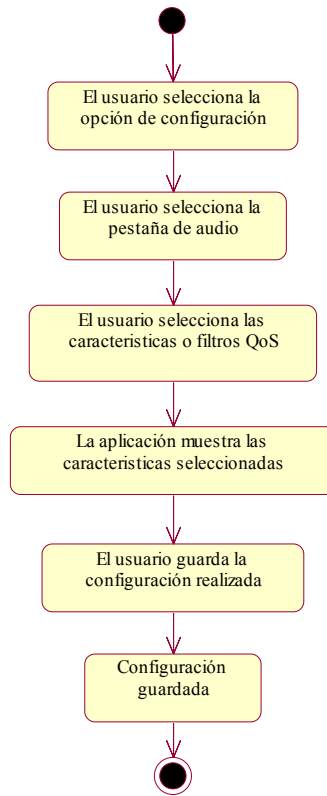


Figura C.8. Diagrama de actividades del caso de uso Configurar cuenta.

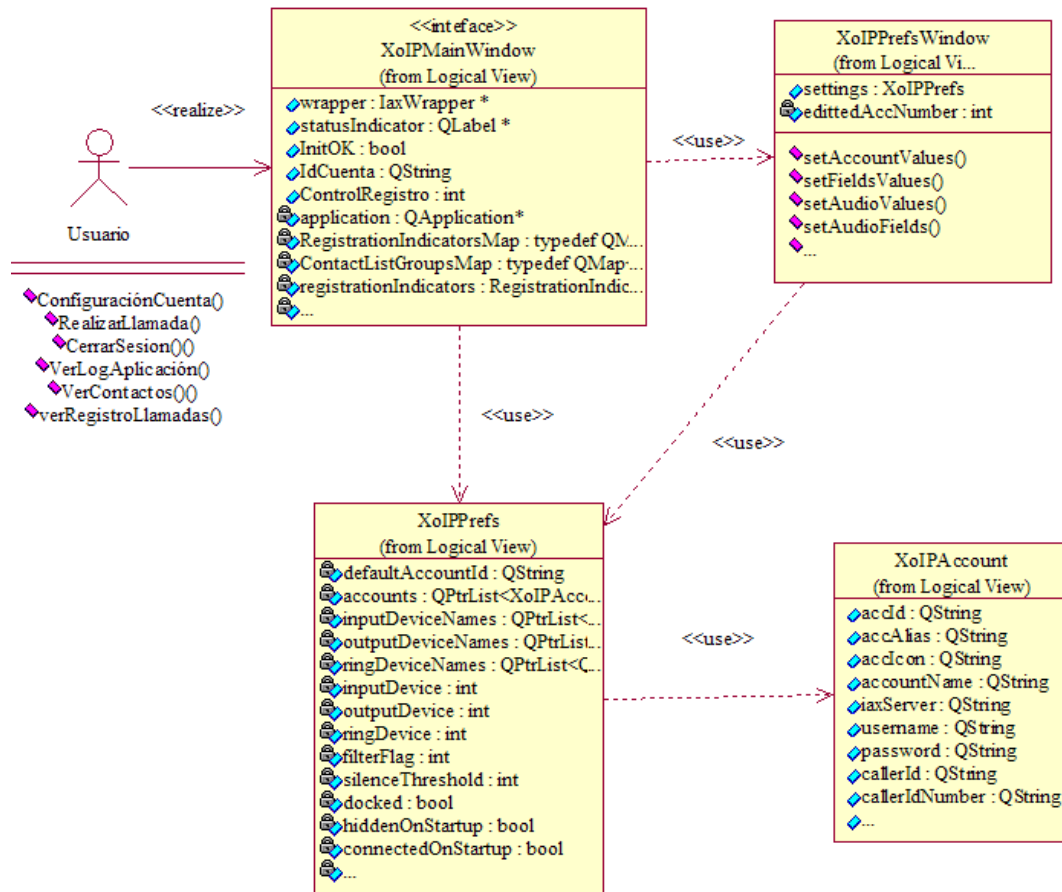


Figura C.9. Diagrama de clases del caso de uso Configurar cuenta.

C.4. Especificación de caso de uso *Log-softphone*

Este caso de uso permite al desarrollador de la aplicación ver todos los mensajes que ocurrieron durante la ejecución de cierta función en la aplicación o en dado caso ver todos los mensajes correspondientes a la ejecución en general de la aplicación. El objetivo es mostrar mensajes que guíen al desarrollador y así conocer que está ejecutando la aplicación. Se tiene al usuario como único actor.

La Tabla C.7 muestra el historial de revisiones y la Tabla C.8 el flujo básico del caso de uso *Log-softphone*.

Tabla C.7. Historial de revisiones del caso de uso *Log-softphone*.

Fecha	Versión	Descripción	Autor
20/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
21/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A.
22/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.8. Flujo básico del caso de uso *Log-softphone*.

Actor	Sistema
1. Este caso de uso inicia cuando el desarrollador o usuario de la aplicación desea ver qué está sucediendo cada vez que ejecuta una función de la aplicación o en su caso ver todos los mensajes que ya han sido arrojados por la misma aplicación	
2. El usuario da click en el botón Ventana de registro	3. Muestra todos los mensajes que han sido arrojados por la misma aplicación
4. Finaliza el caso de uso	

Como precondition se tiene que todos los mensajes deben almacenarse en alguna memoria temporal (*buffer*) antes de ser mostrados por la aplicación.

Las Figuras C.10-C.12 muestran los diagramas de actividades, de secuencia y de clases, respectivamente, del caso de uso *Log- softphone*.

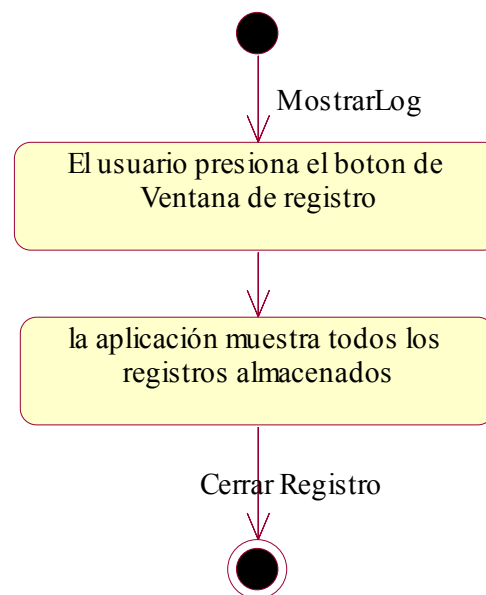


Figura C.10. Diagrama de actividades del caso de uso *Log-softphone*.

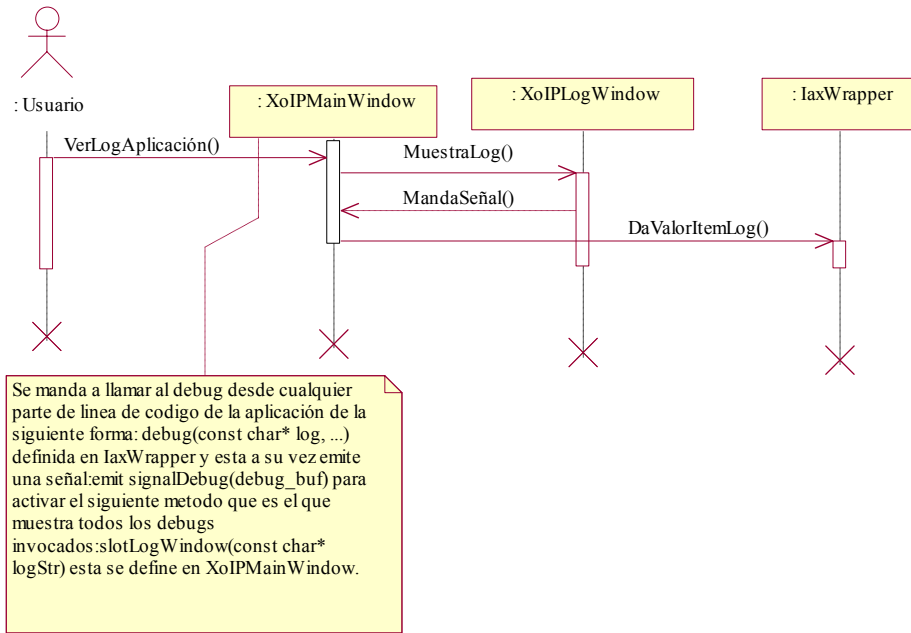


Figura C.11. Diagrama de secuencia del caso de uso Log-softphone.

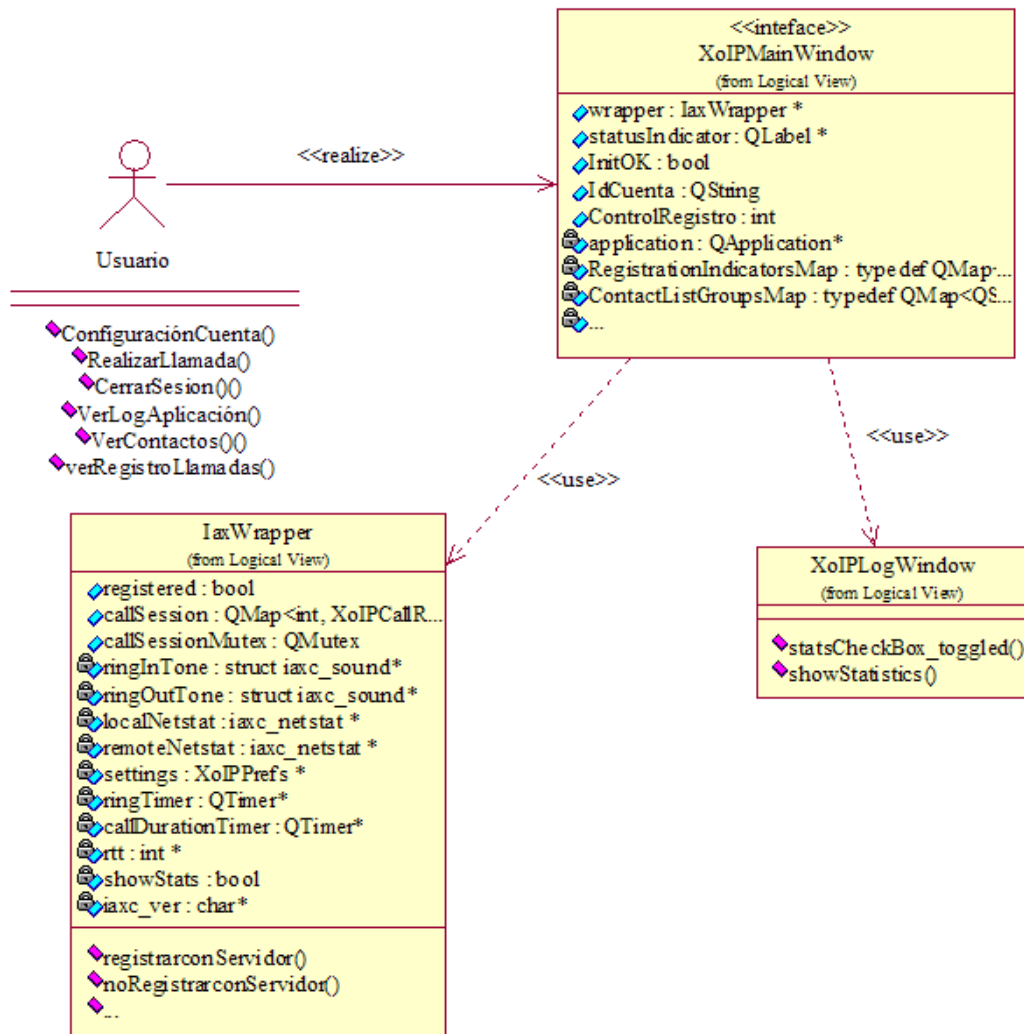


Figura C.12. Diagrama de clases del caso de uso Log-softphone.

C.5. Especificación de caso de uso Cerrar *softphone*

Este caso de uso tiene la finalidad de apagar el teléfono, es decir, se desconecta del servidor y posteriormente cierra la aplicación o *softphone XoIP*. Este caso de uso tiene un sólo actor, el usuario.

La Tabla C.9 muestra el historial de revisiones y la Tabla C.10 el flujo básico de eventos del caso de uso Cerrar *softphone*.

Tabla C.9. Historial de revisiones del caso de uso Cerrar *softphone*.

Fecha	Versión	Descripción	Autor
23/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
24/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A
25/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.10. Flujo básico de caso de uso Cerrar *softphone*.

Actor	Sistema
1. El caso de uso inicia cuando el usuario selecciona la opción Salir	2. Desconecta todas las cuentas que estén registradas en algún servidor VoIP
	3. Apaga o cierra el <i>softphone</i>
4. Finaliza el caso de uso	

Como excepción, si antes de cerrar el *softphone XoIP* no se ha cerrado la sesión de la cuenta por defecto y demás cuentas registradas en su respectivo servidor, el sistema cerrará la sesión de las cuentas.

Lo más recomendable es que antes de que el usuario cierre la aplicación, haya cerrado las sesiones de todas las cuentas que estén registradas en algún servidor VoIP.

Las Figuras C.13-C.15 muestran los diagrama de actividades, de secuencia y de clases, respectivamente, del caso de uso Cerrar *softphone*.



Figura C.13. Diagrama de actividades del caso de uso Cerrar *softphone*.

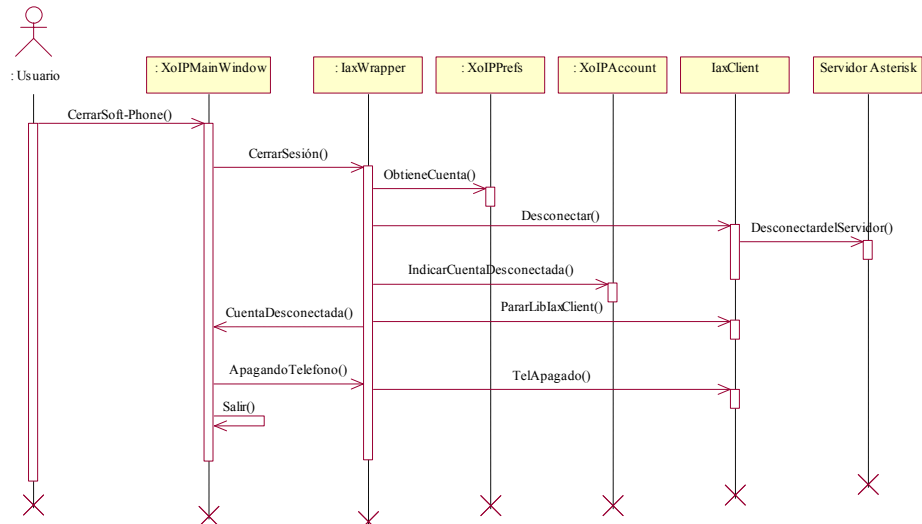


Figura C.14. Diagrama de secuencia del caso de uso Cerrar *softphone*.

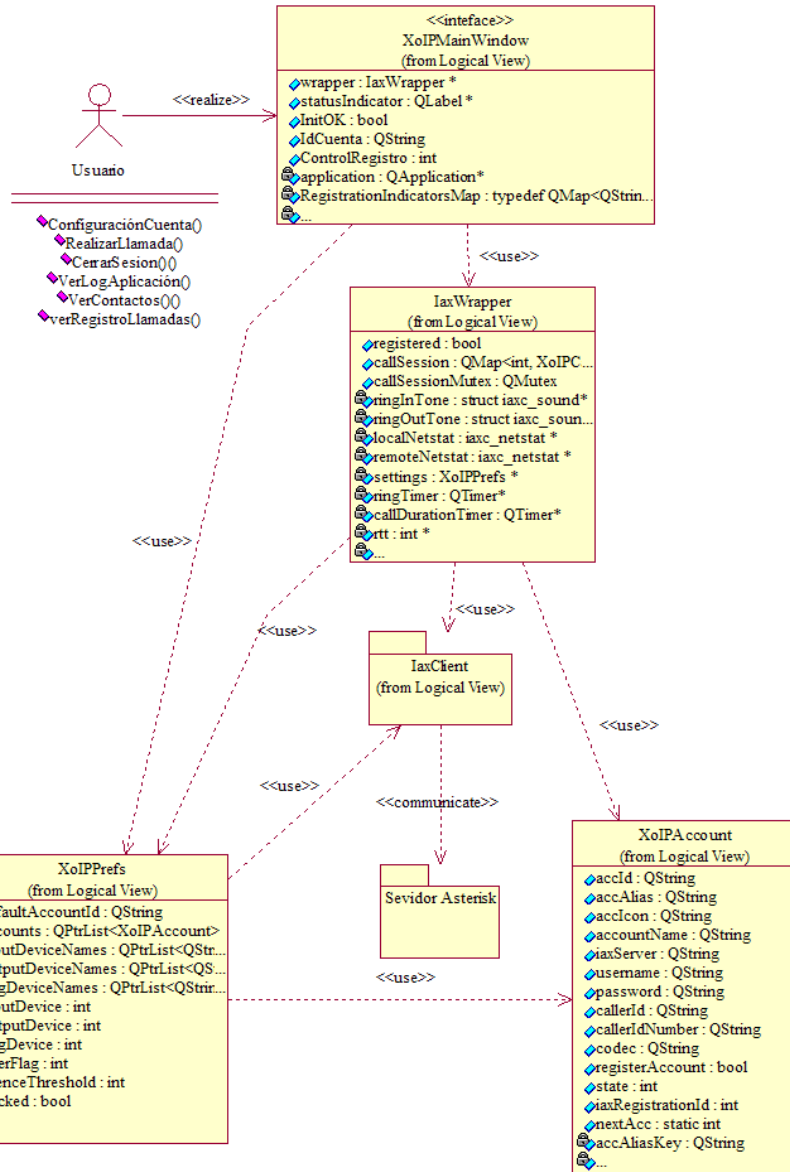


Figura C.15. Diagrama de clases del caso de uso Cerrar *softphone*.

C.6. Especificación de caso de uso Utilizar agenda telefónica

Con el caso de uso Utilizar agenda telefónica el usuario puede ver la lista de contactos que tiene y a través de ella ejecutar las siguientes funciones: agregar, borrar contacto, cambiar información del contacto y realizar llamadas telefónicas. El objetivo de este caso de uso es ver la lista de contactos. En este caso de uso se tiene como único usuario al actor.

La Tabla C.11 muestra el historial de revisiones y la Tabla C.12 el flujo básico del caso de uso Utilizar agenda telefónica.

Tabla C.11. Historial de revisiones del caso de uso Utilizar agenda telefónica.

Fecha	Versión	Descripción	Autor
26/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
27/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A.
28/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.12. Flujo básico del caso de uso Utilizar agenda telefónica.

Actor	Sistema
1. Este caso de uso inicia cuando el usuario decide revisar su lista de contactos o agenda telefónica	
2. El usuario selecciona la pestaña Contactos de la aplicación	3. El sistema muestra la lista de contactos
4. El usuario decide la operación a realizar	5. Ejecuta la petición seleccionada por el usuario
6. Finaliza el caso de uso	

Como precondition, el usuario debe haber iniciado sesión para visualizar la lista de contactos.

Las Figuras C.16-C.18 muestran los diagramas de actividades, de secuencia, y de clases, respectivamente, del caso de uso Utilizar agenda telefónica.

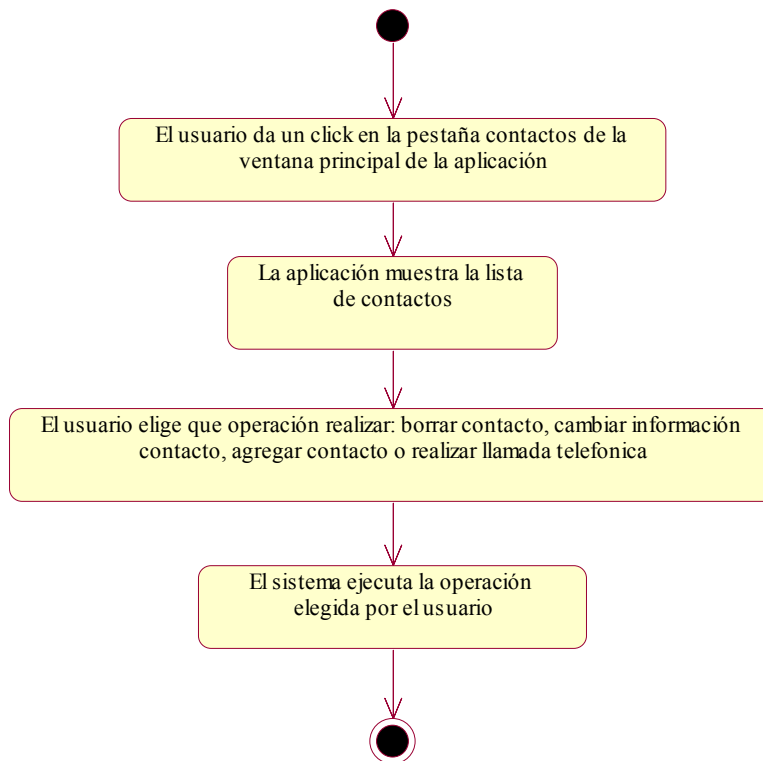


Figura C.16. Diagrama de actividades del caso de uso Utilizar agenda telefónica.

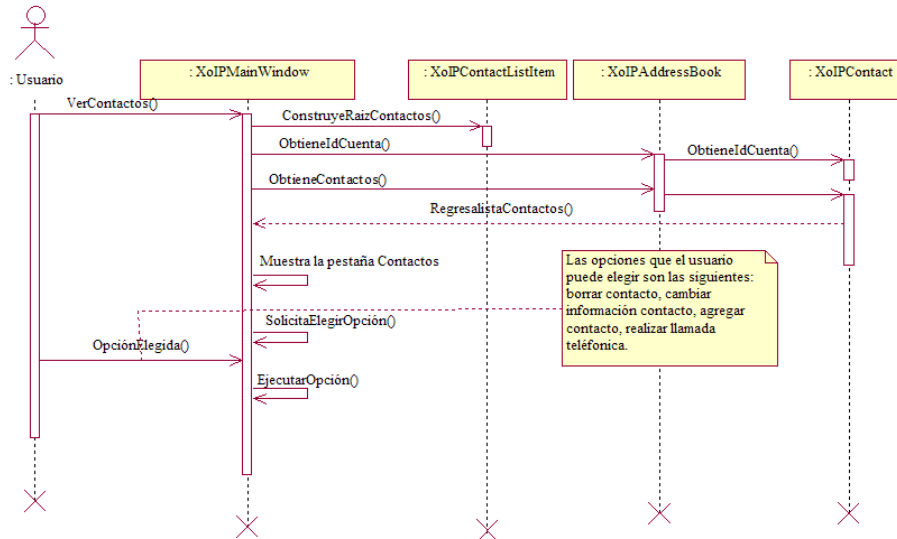


Figura C.17. Diagrama de secuencia del caso de uso Utilizar agenda telefónica.

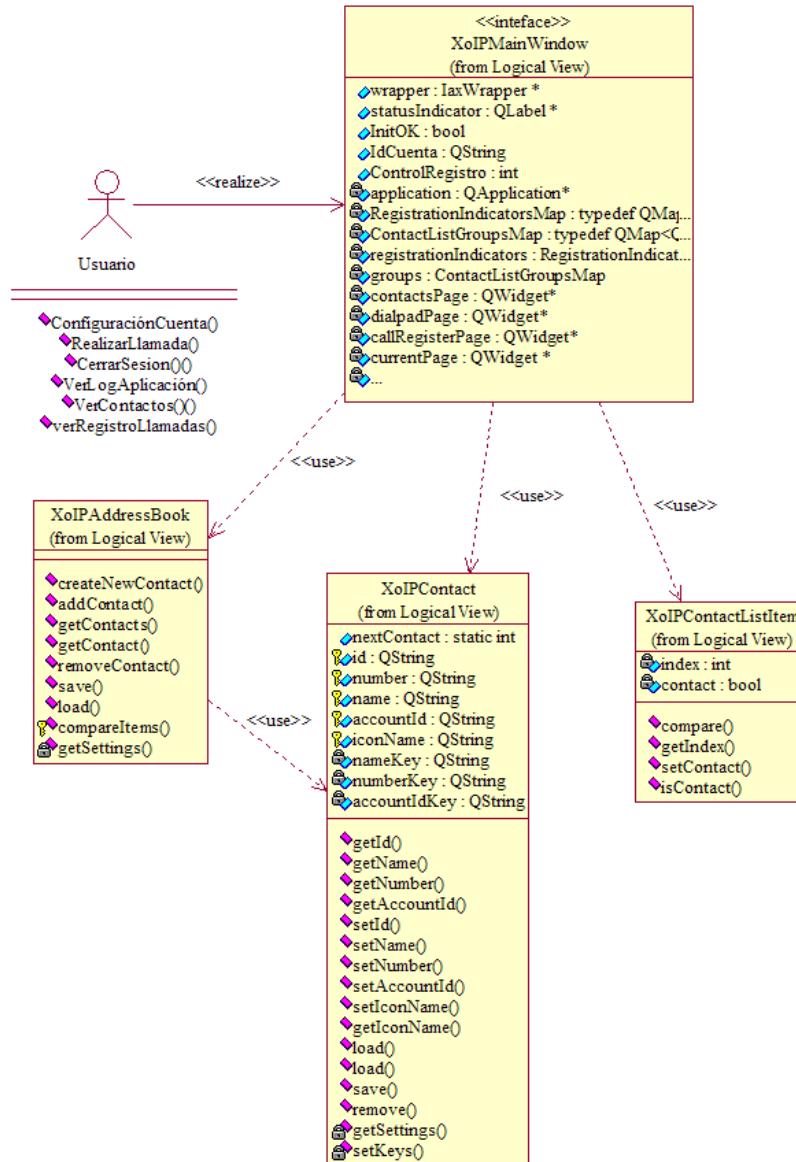


Figura C.18. Diagrama de clases del caso de uso Utilizar agenda telefónica.

C.7. Especificación de caso de uso Recibir llamada telefónica

Con el caso de uso Recibir llamada telefónica, el usuario podrá recibir llamadas telefónicas VoIP de usuarios del mismo Servidor/IP PBX o de algún otro servidor que esté conectado al Servidor/IP PBX. El objetivo de este caso de uso es recibir llamadas VoIP de otros usuarios. Se tiene al usuario como único actor.

La Tabla C.13 muestra el historial de revisiones y la Tabla C.14 el flujo básico de eventos del caso de uso Recibir llamada telefónica.

Tabla C.13. Historial de revisiones del caso de uso Recibir llamada telefónica.

Fecha	Versión	Descripción	Autor
29/Marzo/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
30/Marzo/2006	1.1	Terminación documento	Alberto Lavariega A.
31/Marzo/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.14. Flujo básico del caso de uso Recibir llamada telefónica.

Actor	Sistema
	1. Recibe indicación por parte del Servidor/IP PBX que un usuario registrado está intentando llamar
2. El usuario acepta la llamada que está recibiendo	3. Empieza a recibir los paquetes de voz
	4. Mantiene el establecimiento de la llamada
5. El usuario cuelga y termina la llamada	
6. Finaliza el caso de uso	

Como excepción, si el usuario decide no contestar la llamada, ignora la recepción de la llamada. Si el usuario no quiere contestar puede colgar la llamada.

La precondition existente, el usuario debió haber iniciado sesión para recibir llamadas VoIP.

Las Figuras C.19-C.21 muestran los diagramas de actividades, de secuencia y de clases, respectivamente, del caso de uso Recibir llamada telefónica.

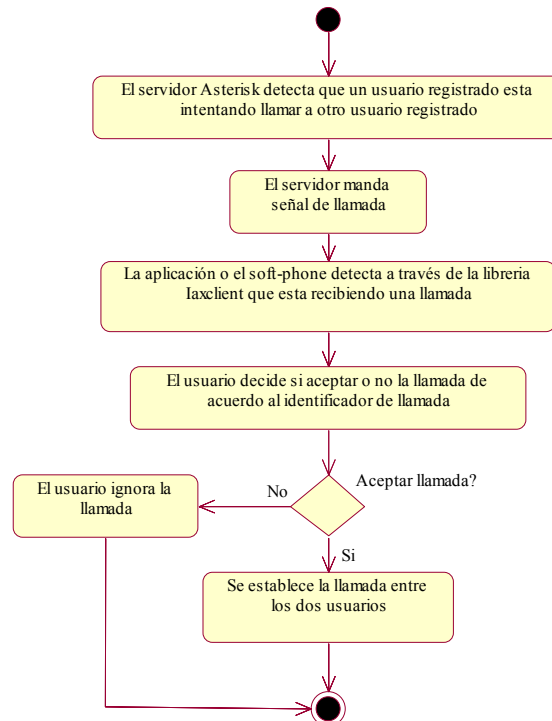


Figura C.19. Diagrama de actividades del caso de uso Recibir llamada telefónica.

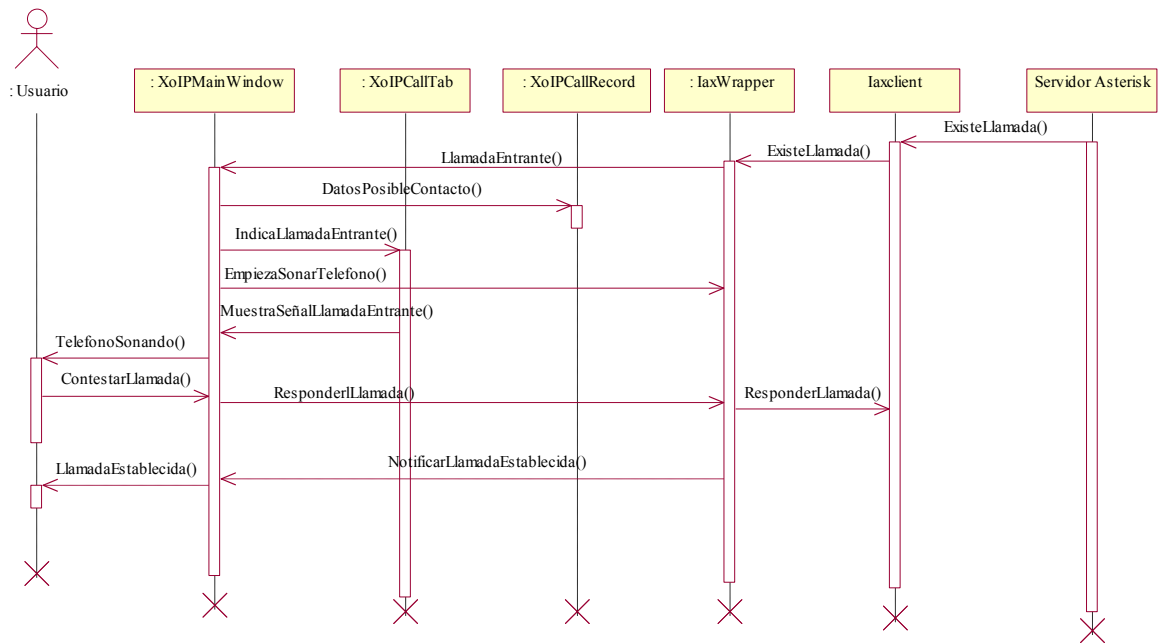


Figura C.20. Diagrama de secuencia del caso de uso Recibir llamada telefónica.

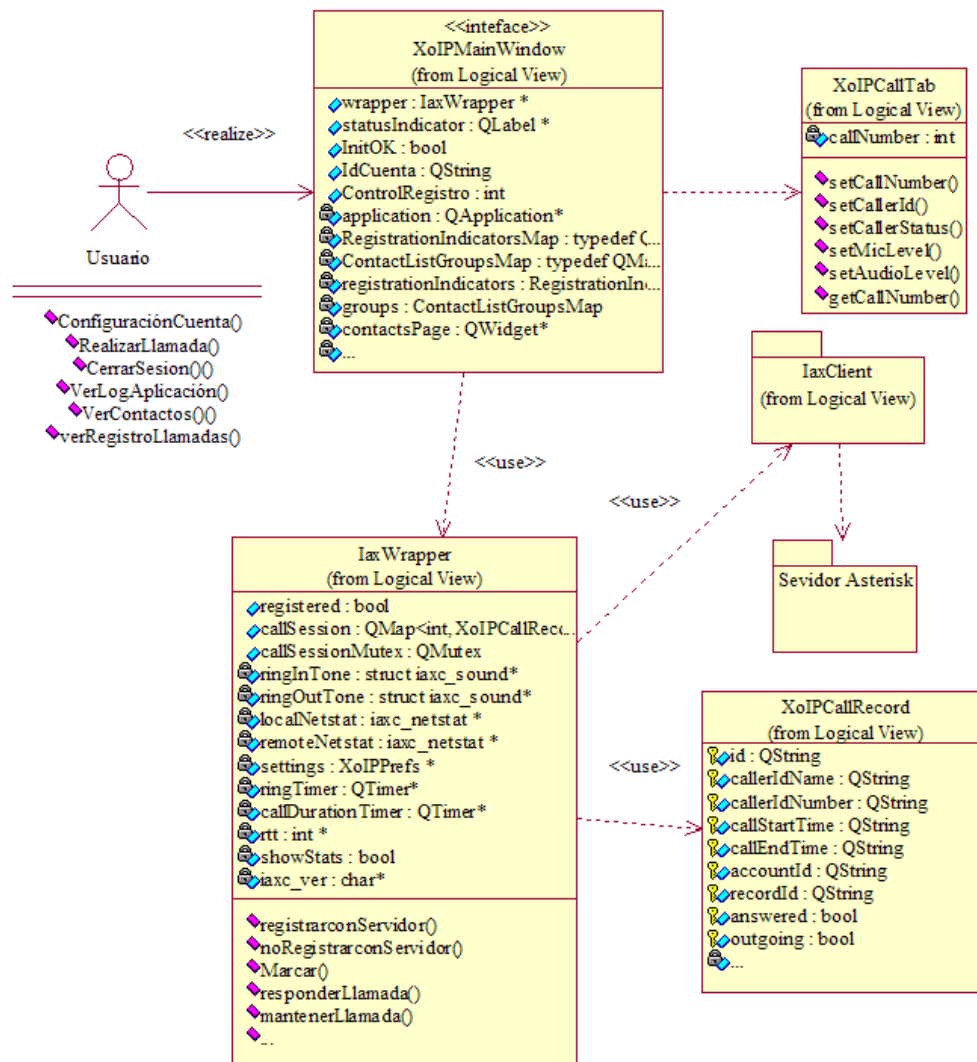


Figura C.21. Diagrama de clases del caso de uso Recibir llamada telefónica.

C.8. Especificación de caso de uso Ver registro de llamadas

Con este caso de uso, el usuario puede visualizar las llamadas que ha recibido, números a los que se ha marcado y las llamadas perdidas. La finalidad de este caso de uso es permitir al usuario visualizar el registro de llamadas. En este caso de uso se tiene al usuario como único actor.

La Tabla C.15 muestra el historial de revisiones y la Tabla C.16 el flujo básico del caso de uso Ver registro de llamadas.

Tabla C.15. Historial de revisiones del caso de uso Ver registro de llamadas.

Fecha	Versión	Descripción	Autor
1/Abril/2006	1.0	Documentación y primeros diagramas	Alberto Lavariega A.
2/Abril/2006	1.1	Terminación documento	Alberto Lavariega A.
3/Abril/2006	1.2	Revisión y corrección de errores	Alberto Lavariega A.

Tabla C.16. Flujo básico del caso de uso Ver registro de llamadas.

Actor	Sistema
1. Este caso de uso inicia cuando el usuario decide ver el registro de llamadas	
2. Selecciona la pestaña Registro	3. Muestra el registro de llamadas
4. Selecciona una función a ejecutar: agregar a agenda telefónica, ver llamadas entrantes, salientes y pérdidas, limpiar registro y realizar llamada	5. Ejecuta la opción elegida por el usuario
6. Finaliza el caso de uso	

Como precondition, el usuario debió haber iniciado una sesión para poder visualizar el registro de llamadas.

Las Figuras C.22-C.24 muestran los diagramas de actividades, de secuencia y de clases, respectivamente, del caso de uso Ver registro de llamadas.

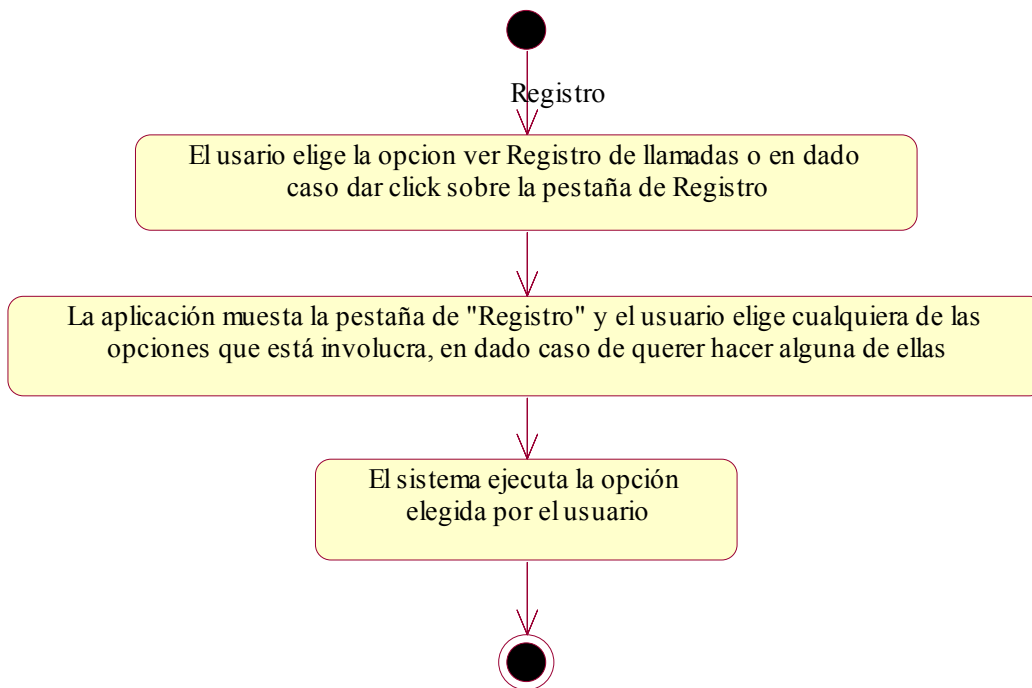


Figura C.22. Diagrama de actividades del caso de uso Ver registro de llamadas.

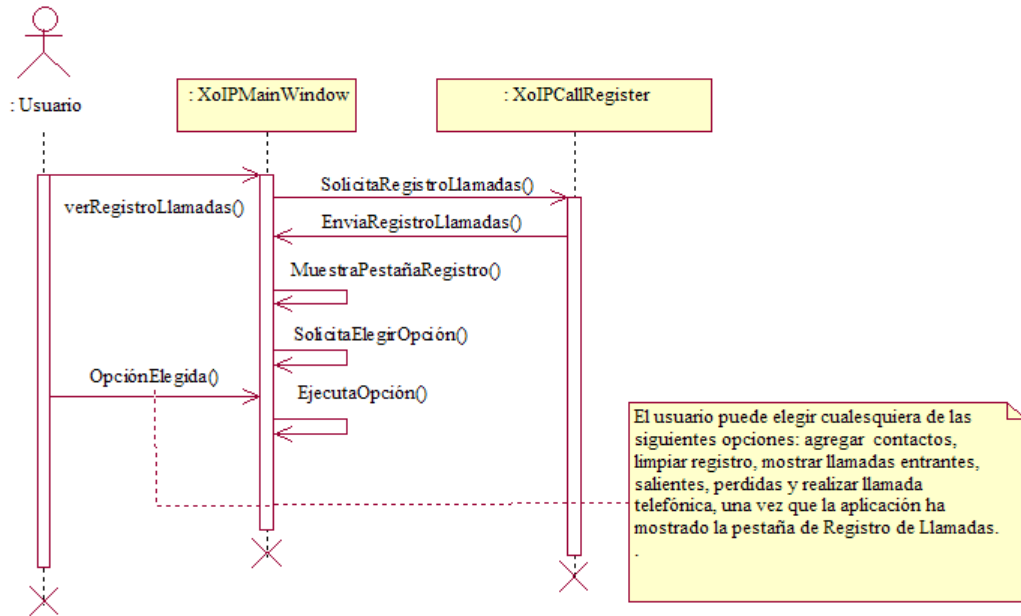


Figura C.23. Diagrama de secuencia del caso de uso Ver registro de llamadas.

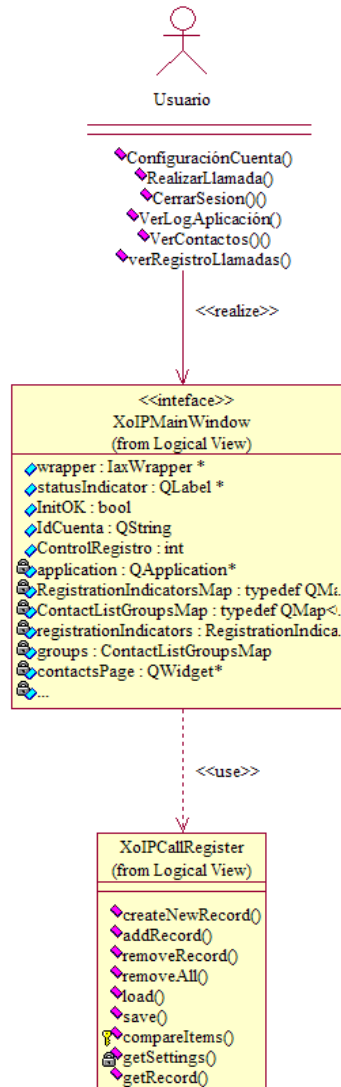


Figura C.24. Diagrama de clases del caso de uso Ver registro de llamadas.

C.9. Identificación de Objetos

La identificación de los objetos es parte importante del diseño de la aplicación, ayuda a establecer claramente el objetivo o funcionalidad de cada objeto. Con esta información los desarrolladores que deseen realizar actualizaciones o modificaciones entenderán el comportamiento de cada objeto dentro de la aplicación.

A continuación se listan las clases y paquetes que conforman la aplicación *softphone XoIP*:

- *XoIPMainWindow*: Esta es la clase principal de la aplicación y se encarga de manipular todos los componentes de la interfaz.
- *XoIPPrefsWindow*: Clase que se encarga de controlar los parámetros de las cuentas de usuario, tipos de dispositivos de audio, filtros seleccionados por el usuario, cargar los valores de las cuentas cada vez que sean requeridos, seleccionar y obtener el codec deseado, borrar y crear cuentas de usuario y guardar todos los cambios de configuración realizados.
- *XoIPPrefs*: Clase utilizada por *XoIPPrefsWindow*. Trabajan conjuntamente en la creación, eliminación de las cuentas de usuario, control de la cuenta por default, cargar los valores de las cuentas, detección de dispositivos de audio, filtros, y todos aquellos datos necesarios que ayuden a la administración de las cuentas.
- *XoIPAccount*: Clase encargada de almacenar, eliminar y leer los datos de conexión e identificación de cuentas: Id, alias, nombre, servidor, usuario, password, identificador de llamada, número de identificación y codec, así como también indicar el estado de registro de la cuenta: inactiva, conectando, aceptada, rechazada.
- *XoIPLogWindow*: Muestra los mensajes arrojados por la aplicación cada vez que se ejecuta una funcionalidad de la misma.
- *IaxWrapper*: Clase que hace uso de las funciones de la librería IAXClient.
- *IAXClient*: Librería de código abierto sobre el protocolo IAX.
- *Servidor Asterisk*: Servidor/IP PBX utilizado para la administración y control de los pares.
- *XoIPContactListItem*: Clase utilizada para el control de los contactos de la agenda telefónica.
- *XoIPAddressBook*: Clase encargada de agregar, eliminar, guardar y cargar la lista de contactos de la agenda telefónica.
- *XoIPContact*: Clase encargada de asignar y obtener los datos de los contactos.
- *XoIPCallTab*: Clase encargada de agregar una pestaña más cuando se está recibiendo o realizando una llamada VoIP. La información que se agrega una vez que la pestaña es creada: el número e identificador de la llamadas, así como el nivel de los dispositivos de audio.
- *XoIPCallRecord*: Clase que tiene como función la de llevar un control sobre las llamadas, es decir, si la llamada fue de entrada, de salida, el tiempo que duró, nombre y número del usuario.
- *XoIPCallRegister*: Clase que trabaja conjuntamente con *XoIPCallRecord* para llevar un registro de las llamadas ya sean de entrada, salida o llamadas perdidas.